A Novel Adaptation Model for E-Learning Recommender Systems Based on Student's Learning Style

Ph.D. Thesis

Shaimaa Mohamed Nafea

Under the Supervision of: **Prof. François Siewe**

Dissertation submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

Software Technology Research Laboratory (STRL)

School of Computer Science and Informatics

Faculty of Computing, Engineering and Media

De Montfort University, Leicester

2019

Declaration of Authorship

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy (Ph.D), at the Software Technology Research Laboratory (STRL) at De Montfort University, United Kingdom.

Neither this material nor its part has been submitted for the award of any other degree or qualification to any other higher educational institution. All the work produced in this thesis was done so in collaboration with my supervisor Prof. **François Siewe.**

This thesis is written by me and produced using LaTeX

Name of candidate: Shaimaa Mohamed Nafea

Signed: ....

Date: ....

Approval

The undersigned certify that they have read and recommended to the graduate school, De Montfort University for acceptance of the thesis entitled "**A Novel Adaptation Model for E-Learning Recommender Systems Based on Student's Learning Style**" submitted by Shaimaa Mohamed Nafea in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

First supervisor:

Prof. François Siewe

Reader in Computer Science

Head of Software Technology Research Laboratory (STRL)

Cyber Technology Institute (CTI)

School of Computer Science and Informatics

De Montfort University

Signed: ....

Date: ....

Second supervisor:

Dr Ying He

Senior Lecturer in Computer Science

Cyber Security Centre (CSC)

Cyber Technology Institute (CTI)

School of Computer Science and Informatics

De Montfort University

Signed: ....

Date: ....

Abstract

In recent years, a substantial increase has been witnessed in the use of online learning resources by learners. However, owing to an information overload, many find it difficult to retrieve appropriate learning resources for meeting learning requirements. Most of the existing systems for e-learning make use of a "one-size-fits-all" approach, thus providing all learners with the same content. Whilst recommender systems have scored notable success in the e-commerce domain, they still suffer from drawbacks in terms of making the right recommendations for learning resources. This can be attributed to the differences among learners' preferences such as varying learning styles, knowledge levels and sequential learning patterns. Hence, to identify the needs of an individual student, e-learning systems that can build profiles of student preferences are required. In addition, changing students' preferences and multidimensional attributes of the course content are not fully considered simultaneously. It is by failing to review these issues that existing recommendation algorithms often give inaccurate recommendations.

This thesis focuses on student learning styles, with the aim of dynamically tailoring the learning process and course content to meet individual needs. The proposed Ubiquitous LEARNing (ULEARN) system is an adaptive e-learning recommender system geared towards providing a personalised learning environment, which ensures that course learning objects are in line with the learner's adaptive profile. This thesis delivers four main contributions: First, an innovative algorithm which dynamically reduces the number of questions in the Felder-Silverman Learning Styles (FSLSM) questionnaire for the purpose of initialising student profiles has been proposed.

The second contribution comprises examining the accuracy of various similarity metrics so as to select the most suitable similarity measurements for learning objects recommendation algorithm.

The third contribution includes an Enhanced Collaboration Filtering (ECF) algorithm and an Enhanced Content-Based Filtering (ECBF) algorithm, which solves the issues of cold-start and data sparsity inherent to the traditional Collaborative Filtering (CF) and the traditional Content-based Filtering (CBF), respectively.

Moreover, these two new algorithms have been combined to create a new Enhanced Hybrid Filtering (EHF) algorithm that recommends highly accurate personalised learning objects on the basis of the students' learning styles.

The fourth contribution is a new algorithm that tracks patterns of student learning behaviours and dynamically adapts the student learning style accordingly.

The ULEARN recommendation system was implemented with Visual Studio in C++ and Windows Presentation Foundation (WPF) for the development of the Graphical User Interface (GUI). The experimental results revealed that the proposed algorithms have achieved significant improvements in student's profile adaptation and learning objects recommendation in contrast with strong benchmark models. Further findings from experiments indicated that ULEARN can provide relevant learning object recommendations based on students' learning styles with the overall students' satisfaction at almost 90%. Furthermore, the results showed that the proposed system is capable of mitigating the problems data sparsity and cold-start, thereby improving the accuracy and reliability of recommendation of the learning object. All in all, the ULEARN system is competent enough to support educational institutions in recommending personalised course content, improving students' performance as well as promoting student engagement.

Keywords: Adaptive learning, Algorithms, Behaviour patterns, Collaborative Filtering, Content-Based Filtering, E-learning, Felder-Silverman Learning Style Model, Hybrid Filtering, k-means clustering, Learning object profile, Learning style, Questionnaire, Rating prediction, Recommendation system, Similarity metric, Student profile.

Dedication

To My Beloved Father

Mr. Mohamed Nafea

Who encouraged me to do my PhD. Thank you for your endless support.

To My Loving Mother

Mrs. Seham Anwar

For her endless love, support, encouragement, and ceaseless prayers.

To My Beloved Brothers

Nafea & Ali

Thank you for being with me during the tough times and for endless love.

To My Sister in Law

Yasmin

I achieved my dream with your support and supplications. I love all of you, and I promise to do my best to make all of you happy.

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful

'and say:'My Lord', Increase me in knowledge' [The Quran 20:114]

I am thankful to Allah Subhanahu wa-ta'ala that by His grace and bounty, I am able to complete my PhD.

I owe a debt of gratitude to several people who provided me with their continual support throughout this arduous journey.

First and foremost, I would like to express my deepest gratitude to my doctoral supervisor **Prof. François Siewe**, who made this research possible. Without his careful instruction, I would never achieve so much success and finish the degree. He was always ready to help and give a piece of good advice. He has been an exemplary guide, critic, and reviewer who instilled in me a deep passion to pursue research. **Prof. Siewe** is a very patient, supportive and attentive supervisor. Whenever I needed his assistance, he was always there. Thanks for believing in me and teaching me. I am grateful to have been able to finish my PhD under his supervision.

In addition, I thank **Dr Ying He** for her input as part of my dissertation committee. My sincere thanks also to **Dr Leandros Maglaras** for his invaluable professional advice.

In addition, I thank my directors and colleagues at the Arab Academy for Science, Technology and Maritime Transport. In particular, I am grateful to **Dr Tarek Abdeen**, **Dr Nermeen Magdi**, **Dr Ahmed Farouk** and **Dr Nihal Ayad** for supporting me in all imaginable ways during my research.

I would like to thank my amazing family for all the support, and constant encouragement they have showered on me over the years, specially my mum **Seham Anwar** and dad **Mohamed Nafea**, for their unconditional love, belief and understanding. Heartfelt thanks to my brothers **Nafea** and **Ali** for their continuous support. Thank you all for encouraging me to climb this ladder, and not look at 'how far, but how well'. Your supports and motivation gave me hope and energy to move on and overcome the challenges that I faced throughout the PhD programme. Rather than give up, you all let me see reasons to move on.

I acknowledge the moral support of all my friends and colleagues, **Shafiaa Rafeah**, **Dr Adekunle Isaac**, **Judith Willems**, **Maria Miranda**, and **Ruba JadAllah**, my sincere thanks to you all.

Lastly, I would like to thank everyone who supported me throughout my PhD who I have not been able to mention here.

List of Publications

Publications arising from this thesis:

1. Nafea, Shaimaa M., François Siewe, and Ying He. "On Recommendation of Learning Objects using Felder-Silverman Learning Style Model." IEEE Access 7 (2019): 163034-163048. DOI: 10.1109/ACCESS.2019.2935417.

2. Nafea, Shaimaa M., François Siewe, and Ying He. "A novel algorithm for dynamic student profile adaptation based on learning styles." In Proceedings of SAI Intelligent Systems Conference, pp. 29-51. Springer, Cham, 2019.

3. Nafea, Shaimaa M., François Siewe, and Ying He. "A Novel Algorithm for Course Learning Object Recommendation Based on Student Learning Styles." In 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), pp. 192-201. IEEE, 2019.

4. Nafea, Shaimaa M., François Siewe, and Ying He. "ULEARN: Personalised Learner's Profile Based on Dynamic Learning Style Questionnaire" In Proceedings of SAI Intelligent Systems Conference, pp. 1105-1124. Springer, Cham, 2018.

5. Nafea, Shaimaa M., Francois Siewe, and Ying He. "ULEARN: Personalized Course Learning Objects Based on Hybrid Recommendation Approach," International Journal of Information and Education Technology vol. 8, no. 12, pp. 842-847, 2018.

6. Nafea, Shaimaa M., Francois Siewe, and Ying He. "An Adaptive Learning Ontological Framework Based on Learning Styles and Teaching Strategies." International Conference on Education and E-learning, 2017.

7. Nafea, Shaimaa, Leandros A. Maglaras, Francois Siewe, Richard Smith, and Helge Janicke. "Personalized Students' Profile Based On Ontology and Rule-based Reasoning."EAI Endorsed Transactions on e-Learning (2016). DOI: 10.4108/eai.2-12-2016.151720.

8. Nafea, Shaimaa M., Leandros A. Maglaras, Francois Siewe, and M. Elemam Shehab. "A novel adaptive learning management system using ontology." In 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 328-335. IEEE, 2015.

List of Abbreviations

Abbreviations	Explanation
AAST	**A**rab **A**cademy for **S**cience and **T**echnology
AEH	**A**daptive **E**ducational **H**ypermedia
AERSs	**A**daptive **E**-Learning **R**ecommender **S**ystems
CBF	**C**ontent-**B**ased **F**iltering
CF	**C**ollaborative **F**iltering
CMS	**C**ourse/**C**ontent **M**anagement **S**ystems
DSP Module	**D**ynamic **S**tudent **P**rofile adaptation **M**odule
ECBF	**E**nhanced **C**ontent-**B**ased **F**iltering
ECF	**E**nhanced **C**ollaborative **F**iltering
EHF	**E**nhanced **H**ybrid **F**iltering
ELS	**E**-**L**earner **S**atisfaction
FSLSM	**F**elder-**S**ilverman **L**earning **S**tyle **M**odel
GUI	**G**raphical **U**ser **I**nterface
HF	**H**ybrid **F**iltering
KB	**K**nowledge-**B**ased
LMS	**L**earning **M**anagement **S**ystem
LO	**L**earning **O**bject
LO Module	**L**earning **O**bjects recommendation **M**odule

LS	**L**earning **S**tyle
LS Module	**L**earning **S**tyles initialisation **M**odule
MAE	**M**ean **A**bsolute **E**rror
MBTI Model	**M**yers-**B**riggs **T**ype **I**ndicator **M**odel
MOODLE	**M**odular **O**bject **O**riented **D**ynamic **L**earning **E**nvironment
OWL	**W**eb **O**ntology **L**anguage
RMSE	**R**oot **M**ean **S**quare **E**rror
RS	**R**ecommender **S**ystem
SMS	**S**tudent **M**anagement **S**ystem
SP	**S**tudent **P**rofile
SQL	**S**tructured **Q**uery **L**anguage
SUS	**S**ystem **U**sability **S**cale
ULEARN	**U**biquitous **LEARN**ing system
VARK Model	**V**isual, **A**ural, **R**ead/write, and **K**inesthetic **M**odel
VLE	**V**irtual **L**earning **E**nvironment
WPF	**W**indows **P**resentation **F**oundation
WWW	**W**orld **W**ide **W**eb
XAML	e**X**tensible **A**pplication **M**arkup **L**anguage
XML	e**X**tensible **M**arkup **L**anguage

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The World Wide Web has substantially grown in the past few decades to become a prominent platform for information and learning. Nevertheless, this massive volume of knowledge available from the Internet has resulted in an information overload making it difficult for users to navigate through and find relevant information. To overcome this issue, personalisation is a broadly used concept which promotes a tailored support system that supports learners to learn [8–10].

However, learners do not have adequate time to manage enormous recommendations, which may sometimes result in lower student satisfaction. As a result, since e-learning systems are not considered useful by students, many of them have moved away from making use of these systems [11–14]. The reason for not finding e-learning frameworks useful is linked to the fact that these learning environments aren't usually able to interact with students including through traditional face-to-face interactions. Another cause is linked to the freedom of learners, where learners are in charge of making few choices for themselves including 'how to learn', 'with whom to learn', and 'which learning route to follow' among others.

Furthermore, in e-learning recommendation systems, adaptability and flexibility of recommendations are crucial due to reasons linked to the ever-changing needs and preferences of students in addition to the continuing change in the functionality of certain learning resources for active students. The Student Profile (SP) is important for building an efficient personalised learning environment [15]. If SPs contain only static information, this will eventually restrict the process of personalisation and suggest irrelevant course content over time. The SP model includes the characteristics of the learner, such as knowledge level and learning styles, both of which are essential in terms of the adaptation mechanism [5, 16, 17].

An adaptive learning environment's critical problem is to discover an appropriate way to dynamically model student preferences and track their changes. Personalisation in e-learning requires changing content and delivering it in line with individual student preferences such as Learning Styles. However, limitations of present e-learning recommendation systems, such as CS383 [10], eTeacher [18] and LSAS [19], do not detect changes in learner preferences due to their respective behaviour patterns. Furthermore, these systems permit the creation of student profiles just once in the beginning without the possibility of future updates [20].

Learning Style (LS) is one of the most significant parameters that consider individual differences while designing personalised learning environments [21, 22]. Several studies have shown that personalised e-learning environments based on particular LSs generate higher rates of student satisfaction, decreased learning times and an increased sense of academic achievement in addition to being more productive [2, 19, 21, 23, 24]. It is also argued that students who exhibit strong preferences for a particular LS, experience learning difficulties when the learning environment fails to support their respective learning style [19, 25, 26].

Recommender Systems (RSs) offer a promising approach to information filtering by helping users to find the most suitable product that matches with their preferences [27–30]. The task of providing personalised content is often described as a recommendation process where users are offered different items and services in line with their needs and preferences [31].

RSs have been used to provide recommendations in diverse industries ranging from e-commerce [9,32,33] to news [34–36], movies [37] and music [37]. Content-Based Filtering (CBF) and Collaborative Filtering (CF) techniques constitute the most popular recommendation systems among others. The CF approach is known to work on the basis of recommending items to a user in line with items which have already been of interest to like-minded users. On the other hand, the CBF approach recommends items similar to those which the target user previously liked. The two above-described approaches expect users to provide a large amount of rating data to show how much users like the items they previously owned before the system provides them with new personal recommendations.

An e-learning recommendation system is responsible for the recommendation of relevant learning materials for various learners in order to assist with improving their performance and achievements [38]. Recommender systems in the field of e-learning constitute a branch of information retrieval where the processing, filtering and recommendation of learning resources to relevant learners takes place in a personalised manner [39]. More specifically, for various reasons, the development of an effective system of e-learning recommendations differs from recommendation in other areas.

First, adaptive e-learning recommendation systems cannot recommend a learning resource to a learner merely because the learning resource was previously liked by other learners with similar preferences. However, the learning styles of the learner must be considered before making the recommendation. Second, learning resources unlike items in other forms of recommendation usually do not have a collection of clearly defined features.

As an example, let us consider a situation where two learners with distinct profiles possess the same rating on the same learning objects. Evidently, under an entirely personalised perspective, the top-n Learning Objects (LOs) list built on predicted ratings will not be similar for both learners. Consequently, it is necessary to consider student learning styles so as to improve the accuracy of the recommendation process.

The remainder of this chapter is organised in a logical way as follows. The problem statement of this research is described in Section 1.2, while Section 1.3 presents the research questions followed by Section 1.4 which describes the aim and objectives of this research. Next, Section 1.5 presents contributions related to this study. Finally, the research methodology and the structure of the thesis are presented in Sections 1.6 and 1.7, respectively.

1.2 Problem Identification

Prior studies on SP adaptation and Learning Objects recommendation have many drawbacks that affect their accuracy [40–43]. In this respect, the key challenges are listed below.

- Firstly, many traditional recommendation approaches developed for e-commerce applications fail to include all learning environment requirements [44,45] and particularly do not take in account the learning process in their recommendation approach [43]. Within a certain system, it is necessary that every user rate a definite number of items in order for the system to be able to make precise recommendations by learning user preferences [46]. In the process of continuous learning, students do not actively rate or comment since they aim at achieving goals within the scheduled time period. Therefore, learners' learning profiles quite often seem isolated from one another. This utmost data sparsity caused due to the above-described factors poses a prominent challenge to the realisation of traditional recommendations techniques [47].

- Secondly, e-learning frameworks generally make use of a “one-size-fits-all” approach where all students are provided with similar learning materials by taking into account only common aspects, and not considering various learning styles and preferences of students. Yet, it is important to note that students vary greatly in terms of their Learning Styles, knowledge levels, background and goals [48–50]. Nevertheless, some personalised systems provide students with the ability to choose content that match their preferences. Other systems construct algorithms based on fixed learning styles that cannot be changed during the student’s progress. These systems provide a somewhat inflexible form of adaptation to adapt to an individual student’s needs in the beginning of a course but does not permit changes afterwards.

- Thirdly, attributes of the learning object profile which include learning styles are crucial in order to provide quality as well as accurate recommendations and have to be taken into account. Hence, this research will look into multi-dimensional attributes of learning objects to personalise a learner’s learning path by means of selection and sequencing suitable learning object profiles according to individual student’s preferences.

- Fourthly, a majority of the current adaptive recommendation techniques suffer from clustering approaches which are of low accuracy. This is due to the high risk of creating clusters that include data points that are actually not too close as well as the risk of obtaining distinct clusters when running the same clustering algorithm repeatedly, denoting that the clustering algorithms do not consistently group the nearest data points.

- Furthermore, there exists an uncertainty in identifying students’ learning styles, particularly with developing and implementing frameworks to effectively infer preferences from students’ actions [26, 51, 52]. It takes a long time while obtaining student behaviour patterns in online learning and at times, these patterns are not strong enough to be used in algorithms [40] [41].

To overcome the above-mentioned drawbacks and substantially enhance the accuracy of top-n learning objects recommendation and student profile adaptation. In this research, a novel e-learning recommendation system, which is called Ubiquitous LEARNing system (ULEARN) will be proposed. First, the Felder-Silverman (FSLSM) learning style questionnaire [2] which determines a learner’s initial learning style is made use of to initialise the learner profile. Next, this student profile (learning style) is dynamically adapted during system usage based on student learning behaviour patterns (i.e. interactions with the system). After that, a novel recommendation algorithm is applied to recommend the most suitable learning objects taking into considerations various student learning styles.

1.3 Research questions

As our contribution is overcoming the current challenges in adaptive e-learning recommendation systems such as information overload, recommendation accuracy, and profile adaptation. Therefore, the main research question addressed by this research is:

How can e-learning adapt student profile dynamically in order to recommend the most suitable learning objects?

The primary aim of this study constitutes addressing the question above; it is helpful to first split it into several smaller problems that will be addressed. The idea is that the individual solutions to each of these sub-questions can be put together to form a complete solution to the main question.

RQ1. What is an appropriate architecture for an adaptive learning object recommender system?

RQ2. What is the best way to initialise student learning styles?

RQ3. What similarity metrics should be used in the learning object recommendation algorithm to achieve the best accuracy?

RQ4. How can personalised learning objects be recommended based on student's learning styles and object profiles?

RQ5. How to model dynamic student profiles based on student learning behaviour patterns?

Addressing all of these questions satisfactorily still comprises an open problem within the field of adaptive e-learning recommendation systems. Although many studies in the past [53, 54] have attempted to address some of these issues independently, no generic adaptive e-learning system has been developed to address all of these issues in an integrative and effective way. The following section will demonstrate research aims and objectives in details.

1.4 Aims and objectives

The primary aim of this research constitutes improving the effectiveness of e-learning recommender systems by dynamically tracking changes in student learning styles based on their learning behaviour patterns to recommend suitable learning objects. In order to accomplish the proposed aim, the objectives of this study are as below:

1. To carry out a critical literature survey related to the adaptation of e-learning recommendation systems as well as to identify research challenges and gaps (Chapter 2).

2. To develop a novel e-learning architecture to provide personalised learning objects recommendations according to individual student's learning styles (Chapter 3).

3. To develop an algorithm for initialisation of student profile (learning styles) (Chapter 4).

4. To conduct an experimental study in order to determine the best similarity metrics for the purpose of adaptive e-learning from the pool of available similarity metrics (Chapter 5).

5. To develop an efficient personalised learning objects recommendation algorithm by taking into account student learning styles (Chapter 5).

6. To develop an adaptation algorithm for updating students' profile dynamically according to student learning behaviour patterns (Chapter 6).

7. To implement a prototype adaptive learning object recommender system and evaluate its performance in real-world scenarios (Chapter 7).

Figure 1.1 presents the linkages between Research Questions (RQs) and Research Objectives (ROs) to achieve research aims. The following section describes the specific contributions of this research in more detail.

1.5 Contributions

This thesis introduces a novel algorithm for adaptive e-learning recommender system, known as the ULEARN (Ubiquitous LEARNing) system. The idea behind the development of the ULEARN recommender system is to personalise the recommendation of learning objects in line with adaptive student's profile (learning styles). The main contributions of this thesis are fourfold.

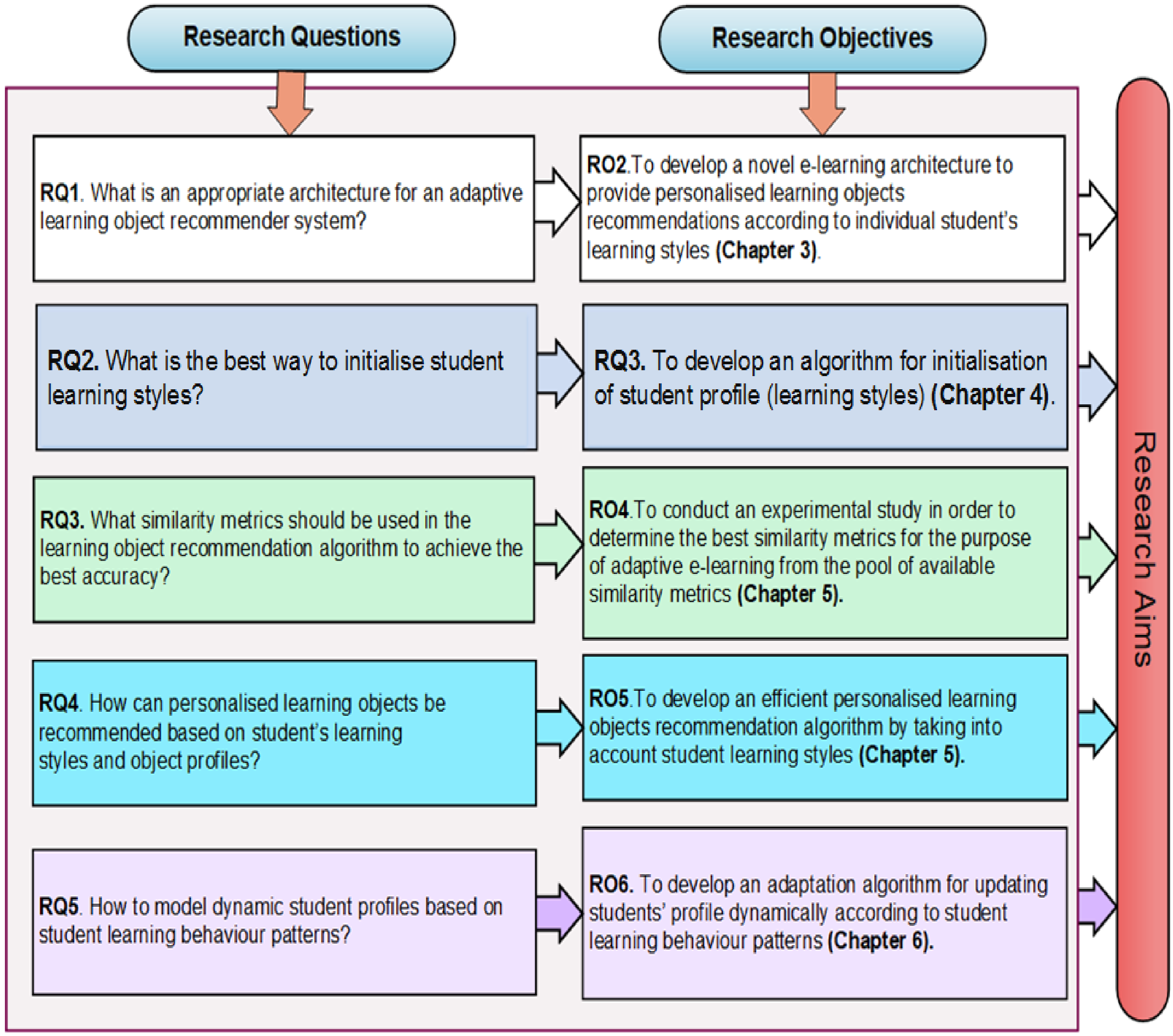


Figure 1.1: Research questions and research objectives to achieve the research aim

- *(C1)* **A novel student learning style initialisation model**. An innovative algorithm in order to construct an adaptive learner profile in line with the FSLSM during the registration process called "Dynamic ILS Questionnaire" is proposed in this thesis. Initially, an empirical study was carried out to ascertain the sequence of questions for all of the four dimensions within the FSLSM model followed by the construction of the algorithm in line with the questions so as to compute the initial learning style of the learner. This algorithm possesses another novel feature which is its ability to deduce learners' learning style from their response to a few questions in each of the four dimensions of the questionnaire, thus saving significant time and effort from having to answer all 44 questions within the FSLSM model. Later within this research, specifically in Chapter 4 comprises the introduction to our new algorithm.

- *(C2)* **Evaluation of similarity metrics performance in the context of learning objects recommendation**. An experimental study was carried out to determine the most appropriate similarity metrics to utilise in an adaptive recommender algorithm for learning objects. The experimental study will later be introduced in Chapter 5.

- *(C3)* **A novel recommendation model**. An innovative algorithm to recommend learning objects in line with student learning styles and the learning object profiles is proposed. In addition to improving the accuracy of recommendations, the new algorithm overcomes the cold-start problem and the data sparsity problem. This contribution will be discussed in Chapter 5.

- *(C4)* **A novel algorithm for adapting student learning styles dynamically**. Considering learning behaviour patterns such as time spent on learning objects and number of messages exchanged, an efficient algorithm has been proposed to dynamically adapt student learning styles. This includes the proposal of a new methodology to transform learning behaviour patterns into the FSLSM learning style preferences in order to update student profiles dynamically. More about this module will be discussed in Chapter 6.

1.6 Research Methodology

This section comprises the method of research as applied in this thesis linking new knowledge developed as part of this research to the process leading to outcomes. For purposes of scientific research within all domains of Computer Science and Software Engineering, the majority uses the constructive research method [55, 56]. The constructive research aims at producing novel solutions to practical or theoretical problems through construction of models, algorithms, diagrams or systems. The constructive research method is appropriate for this thesis given the aims and objectives of the thesis stated above to develop a novel adaptive e-learning recommender system based on student learning styles. As customary in the constructive approach, the research method comprises six stages, as follows.

- **Stage 1** ***(Define research problem)***: Find practically relevant problems that also have research potential as previously explained in Section 1.2.

- **Stage 2** ***(Literature review)***: Conduct background research initially with a theoretical literature view to reinforce the understanding of all approaches related to the research question. It produces a critical review of the state-of-the-art approaches in adaptive e-learning and student's learning styles identification; conducts number of comparative studies to demonstrate the advantages of our approach; identifies the research gap related to the development of adaptive e-learning environment, as well as proposes a detailed research methodology to address the research gap. Therefore, this review has become the foundation of the proposed system introduced by this thesis.

- **Stage 3** ***(Construct a solution)***: This phase aims at developing a novel solution for improving the accuracy of e-learning recommendation systems. Based on the investigation in the first phase, an innovative solution to build an adaptive e-learning recommendation system shall be introduced.

- **Stage 4** ***(Implement and refine the solution)***: This stage focuses on developing (i) the algorithm for initialising the student learning styles using the ILS questionnaire; (ii) the algorithm for recommending learning objects in line with student learning styles; and (iii) the algorithm to dynamically adapt student learning styles based on student learning behaviour patterns.

- **Stage 5** ***(Evaluate the performance of the solution)***: This phase aims at assessing proposed ULEARN system performance, in addition, results have been discussed

- **Stage 6** ***(Examine the scope of applicability of the solution)***: The proposed system (ULEARN) will be evaluated using real-world scenarios in order to examine its capability while recommending the most suitable learning objects which meet the student's preferences.

The research conducted in this thesis proposes novel recommendation algorithms for the purpose of e-learning based on students' learning styles. Table 1.1 presents the proposed approach with respect to the thesis chapters.

1.7 Structure of thesis

This section outlines the structure of this thesis, which is ordered into eight chapters, in addition to a summary of the contents within each chapter as described below.

- **Chapter 2: Background and Literature Review.**

  This chapter introduces the initial relevant research background by providing a brief overview of the current approaches related to adaptive e-learning recommendation as well as student profile adaptation and their limitations.

Table 1.1: The constructive research stages with respect to the thesis chapters

Stages of Constructive Research		**Related Chapters**
(Stage 1)	***Define research problem***	-The first phase is addressed in ***Sections 1.1*** and ***1.2*** of the current chapter.
(Stage 2)	***Literature review***	-The second phase is presented in ***Chapter 2***, which contains a literature review.
(Stage 3)	***Construct a solution***	The third phase is discussed in ***Chapters 3***, ***4***, ***5*** and ***6***. - ***Chapters 3*** comprises the ULEARN system architecture. - ***Chapters 4*** presents student profile initialisation. - ***Chapters 5*** presents learning objects recommendation algorithms and an evaluation of similarity metrics performance. - ***Chapters 6*** covers student profile adaptation algorithms based on their learning behaviour patterns.
(Stage 4)	***Implement and refine*** the ***solution***	-The fourth stage which is presented in ***Chapter 7*** covers the implementation of the proposed system (ULEARN) modules.
(Stage 5)	***Evaluate the performance of the solution***	-***Chapter 7*** evaluates the approach in addition to a discussion of the results. The evaluation shows how the objectives of the proposed algorithms are achieved.
(Stage 6)	***Examine the scope of applicability of the solution***	-***Chapter 8*** outlines the contribution, limitations and future directions of this study.

- **Chapter 3: Ubiquitous LEARNing system (ULEARN) Architecture.**

  This chapter presents the architecture of the proposed Ubiquitous LEARNing (ULEARN) approach and its modules such as a new dynamic ILS questionnaire, a new learning objects recommendation module and a novel dynamic student profile adaptation module. This chapter also introduces an illustrative example to clarify how the different modules work together within one system of the ULEARN.

- **Chapter 3: Dynamic Learning Styles Questionnaire for Student Profile Initialisation.**

  This chapter presents a novel proposal of the dynamic ILS Questionnaire module, which is made use of within ULEARN in order to dynamically decrease the number of questions within the FSLSM questionnaire used to initialise learning styles of the learner.

- **Chapter 3: Learning Objects Recommendation.**

  This chapter introduces an innovative and efficient recommender algorithms which recommend personalised learning objects by considering various student learning styles. Also included within this chapter is an experimental study which permits an investigation of suitable similarity metrics for use in an e-learning recommender algorithm followed by a novel algorithm to recommend top-n Learning Objects (LOs) according to student learning styles.

- **Chapter 6: Dynamic Student Profile Adaptation.**

  This chapter introduces the latest algorithms designed to track learning behaviour patterns of students besides identifying learning styles and maintaining dynamic student profiles in a Recommender System (RS).

- **Chapter 7: Implementation and Evaluation of ULEARN Recommender System.**

  This chapter presents the actual implementation of the proposed e-learning recommender system. This chapter also provides a detailed definition of the evaluation criteria as well as the results of implementing ULEARN components which includes measuring student satisfaction using two valid questionnaires E-Learner Satisfaction (ELS) and System Usability Scale (SUS).

- **Chapter 8: Conclusions and Recommendations for Future Work.**

  The main findings of this thesis are summarised within this chapter followed by the discovery of its strengths and weak points and identification of areas which require improvement. Future directions for research in the area related to personalised e-learning recommendation techniques are also suggested.

Chapter 2

Background and Literature Review

Presented in this chapter is the background related to this research in the field of adaptive and personalised e-learning. As introduced earlier, this research combines several research domains: e-learning and learning styles on the one hand together with recommendation techniques and clustering in recommender systems on the other. In addition, relevant sections within this chapter explore how this research relates to previous work undertaken in the above-mentioned area and in what ways it significantly differs from these. The purpose of this chapter aims at addressing research objective 1 (Section 1.4) which is described as follows:

> ***Objective 1:*** To carry out a critical literature survey associated with the adaptation of e-learning recommendation systems, as well as to examine related areas of research that require further exploring, by identifying key challenges as well as gaps found in the literature and focusing on contributing in this area, each of which serve as motivation and background knowledge to support all the research questions defined in Chapter 1 (Section 1.3).

This chapter is categorised into six sections. An overview of learning theories and learning styles, which are crucial for developing adaptive e-learning applications are presented in Sections 2.1 and 2.2 respectively. Next, Section 2.3 seeks to explain the concepts of e-learning and educational technology. Section 2.4 discusses adaptive e-learning in addition the architecture of adaptive systems and an overview of evaluation methods, while Section 2.5 describes recommender systems and their techniques. Subsequently, Section 2.6 presents clustering in recommender systems, followed by Section 2.7 which describes similarity metrics in e-learning systems. Lastly, Section 2.8 comprises a summary of the entire chapter, which has been drawn up to emphasis major gaps found in literature as well as areas which would need to be explored further, thus highlighting the contributions of this research in contrast to the present literature.

2.1 Learning theories

Learning theories seek to explain how learners think and what features control their learning behaviour. These theories which comprise frameworks and are usually conceptual in nature, determine how knowledge is absorbed, processed and retained throughout the process of learning [57]. Owing to the fact that interpretation of the learning process is fairly complex, quite a few learning theories developed in the past century have different viewpoints. Learning theories are vastly essential since design principles for learning environments, materials, and tasks could be derived from these theories in addition to explaining the phenomenon of learning [58].

Furthermore, it is also possible for learning theories to create actionable knowledge from learner actions to be used in instructional and learning design [59]. The three major approaches to learning theories which have had an immense influence on e-learning and instructional design are described below as follows:

- **Behaviourism theory:** Within this behavioural approach, learning is defined by the change in current behaviour or the acquisition of a new behaviour [60]. According to this theory, the learner reacts in response to being triggered by an environmental stimulus [60]. While these responses are usually ordered to simulate desired behaviour, the new behaviour is repeated so as to become automatic. Thus, behavioural changes are generated by the above-described associations between stimuli and responses.

- **Constructivism theory:** As maintained by this theory, learners construct knowledge on the basis of their interaction with the environment [61,62]. As stated in [61], a notion among constructivists is that knowledge is constructed in the minds of humans through basic processes such as discovery and problem solving among others.

- **Cognitivism theory:** This approach which emerged in response to the limitations of the behaviourism theory [61] deals with the learner's information processing habits [61,63,64], particularly on the thought process involved behind certain behaviours. Generally, changes in behaviour serve to indicate the happenings inside the learner's mental model [61].

Table 2.1: Learning theories with respect to learner characteristics

Learning Theories	Learner characteristics	Authors
Cognitivism theory	Learning style	[65–68]
	knowledge Level	[69–73]
	Emotions	[74–78]
Behaviourism theory	Learning effect	[79–81]
Constructivism theory	Learner context (external learning environment)	[82–85]

All the three learning theories described above possess distinct effects related to the development of adaptive e-learning systems since they seek to interpret how effective learning takes place through such systems. Table 2.1 shows learner characteristics used when particular theories are employed. Since the behaviourism theory does not take into account the learner's internal mental processes, this approach is particularly not applicable in terms of implementing adaptive e-learning systems. Additionally, behaviourism does not learn for creative thinking or problem-solving. In fact, students only recall few generic details or automatic responses and not known to take any initiative in order to change or improve things [86].

Since constructivists view the learning process differently, they believe that knowledge is generated by learners through the integration of new information with existing knowledge and experience [87]. In addition, it is thought that constructivism has a rather positive effect in the learning process whereby learners are involved in their respective progress and simultaneously control their own learning while making use of interactive and adaptive instructional techniques and eliminating grades. Consequently, constructivism in education poses a threat since it is known to generate uncontrolled learning instead of a systematic way of learning, thus making evaluation process difficult.

The cognitive approach which seeks to combine artificial intelligence and learning analytics is used to enhance adaptive and personalised learning. This theory is more commonly used in learning systems as demonstrated in Table 2.1.

Hence, this study is focussed on proposing an adaptive e-learning environment which recommends every suitable learning object according to a student's learning styles. These Learning Styles are frequently applied in education, and their influence on students' learning process and outcomes are increasingly emphasised upon [88, 89]. The following section comprises descriptions of numerous Learning Style models.

2.2 Learning Styles

A Learning Style (LS) is denoted by the manner in which a learner absorbs, processes and retains knowledge [90]. These styles are generally different for each student based on personal factors. For instance, during an experiment, some students are able to grasp quite well just by following verbal instructions, whereas others gain a sound understanding only after physically practising the experiment on their own. It is key that educational systems recognise and consider different learning styles so as to improve the learning process for all.

Today's literature contains quite a few definitions of the term LS. According to [90], Learning Styles comprised the means by which information was processed in addition to constituting the learner's strengths as well as preferences. In [91], the authors perceived Learning Styles as the *"composite of characteristic cognitive, affective, and physiological factors that serve as relatively stable indicators of how a learner perceives, interacts with, and responds to the learning environment"*. Subsequently in [92], it was widely understood that Learning Styles were *"based on individual differences in learning in addition to the learner's preference for employing different phases of the learning cycle"*.

As pointed out by [93], Learning Styles constitute a crucial factor during the learning process as per numerous educational researchers who have also suggested that their implementation in the educational system would drastically enhance the learning process. Moreover, [2] speculated that learners accustomed to a specific style of learning could experience difficulties especially in situations where the way of teaching fails to match with the learner's style of learning. According to [93], learners are more acquainted to know themselves better, especially in terms of their strengths and weaknesses once they become aware of their Learning Styles. It is through focusing on these weaknesses that they might develop their learning processes. Furthermore, Learning Styles form a supporting factor throughout the process of designing online learning environments.

For the purpose of classifying learners into specific categories, the development of several learning style models has taken place. Learner types are made use of to enhance the learning potential of a learner by providing relevant learning material. Certain well-known learning styles include models developed by David Kolbs (Kolb's Learning Style) [1], Peter Honey and Alan Mumford (Honey and Mumford's learning style model) [6], Neil Fleming (VARK learning style model) [7], Katherine Briggs and Isabel Myers (Myers-Briggs Type Indicator) [94], and lastly Richard Felder and Linda Silverman (Felder-Silverman Learning Styles model) [2].

The following sub-sections review the above-mentioned Learning Style models and the methods that were used to characterise each learner's preferred styles of learning.

2.2.1 Kolb Learning Style

Developed by David Kolbs [1], this theory is closely related to experiential learning, i.e. knowledge creation when a change of experience occurs in the process of learning. Kolb's LS model is measured using LSs Inventory (LSI) [95]. The Kolb model [1] which is measured using LSs inventory includes four basic learning modes, namely, Concrete Experience (CE), Reflective Observation (RO), Abstract Conceptualization (AC), and Active Experimentation (AE), as shown in Fig. 2.1 [1]:

- ***Concrete Experience (CE)*** includes senses such as hearing, seeing, touching and feeling;
- ***Reflective Observation (RO)*** represents reflecting about the knowledge developed;
- ***Abstract Conceptualisation (AC)*** defines logical examination of the new knowledge;
- ***Active Experimentation (AE)*** refers to the practical application of concepts and ideas.

The sequence of the four modes as defined by Kolb [1] is as follows: The first stage includes an experience that the learner encounters. This experience establishes the foundation for observation. Next, the learner reflects on this experience and interprets its meaning followed by the creation of abstract ideas and concepts based on their interpretation. Lastly, the ideas are tested and applied in the environment. This four-stage process is then repeated where the latest experience is used as the concrete experience of the first mode. The process then cycles back to the initial stage of the experience. In order to completely understand this topic, it is necessary that the learner completes the entire cycle of all four modes.

As part of the Kolb Theory, a learner is positioned at the intersection as illustrated in Fig. 2.1. This intersection occurs where the vertical line (which connects CE and AC) and the horizontal line (which joins RO and AE) connect. Kolb [1] divided Learning Styles into four types:

The *Accommodator (Concrete/Active)* Style where knowledge is gained by learners through the first and last stage, i.e., through concrete experiences and active experimentation. It is also presumed that though such learners might be impatient sometimes, they are generally good problem solvers.

The *Diverger (Concrete/Reflective)*) Style where learners are usually creative and are always thinking of the entire picture. Such people also excel at brainstorming by being able to reflect from different perspectives.

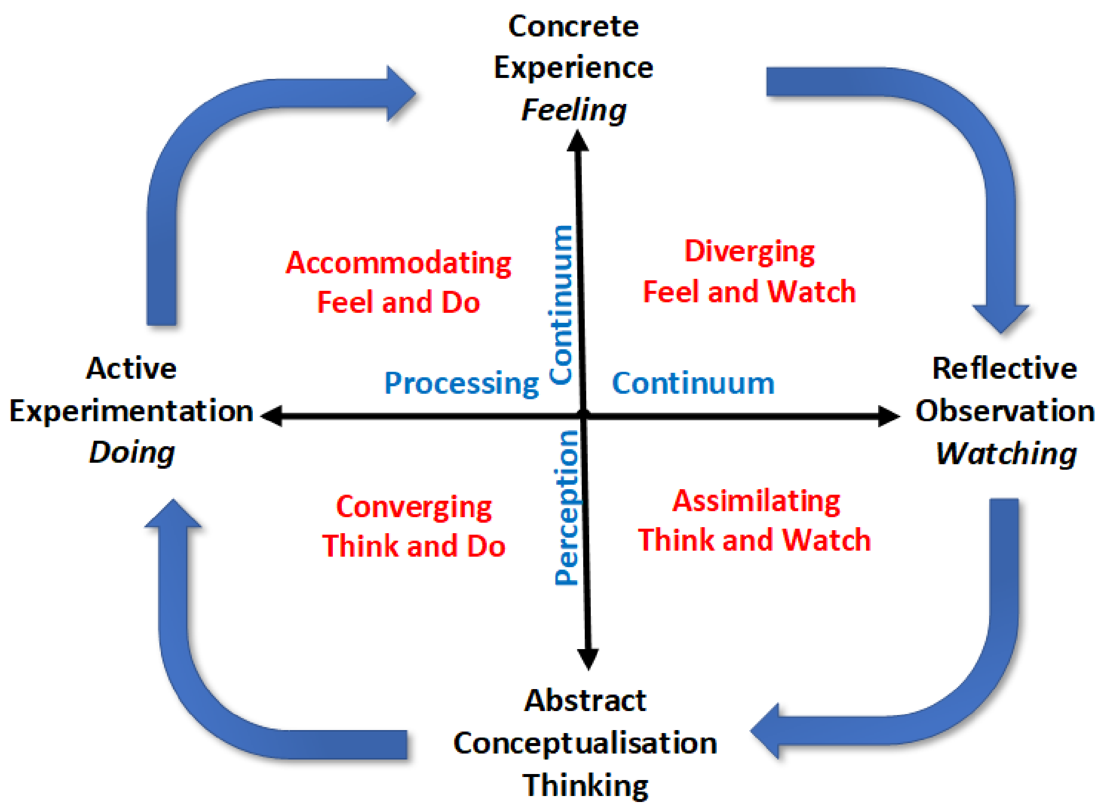


Figure 2.1: Kolb's experiential learning cycle [1]

The *Converger (Abstract/Active)* Style where learners are highly skilled in practically applying their ideas to their surroundings in addition to being creative and excellent decision-makers.

The *Assimilator (Abstract/Reflective)* Style where learners are experts in abstract conceptualisation and reflective experiences. It is known that such learners are highly skilled in creating theoretical models and generally excel in abstract ideas while not being too interested in the application of concepts. In general, these people avoid interactions with people.

Although Kolb's theory has been received quite positively for performance improvements specifically in the educational sector, numerous issues have been observed with the model [96]. It is known that Kolb's model does not justify the significant number of new skills specific to e-learning environments and technology such as social learning and interactive environments [97]. In addition, this model does not possess the characteristic to be able to locate and utilise information which is delivered in several distinct formats, such as audio, video, text, animation and image. It is necessary to note that learners prefer different learning types over time based on various situations. Hence, the LSI may not be adequately reliable to identify style differences among e-learners and plan an e-tutorial on that basis [98].

2.2.2 Honey and Mumford learning style model

Inspired by Kolb's four Learning Styles [1], the Honey and Mumford model was developed by a pair Peter Honey and Alan Mumford [6] to further identify four distinct Learning Styles, namely, *Activist*, *Theorist*, *Pragmatist*, and *Reflector*, characteristics of which are described below in Table 2.2.

Table 2.2: Learning Styles as defined by Honey and Mumford [6]

Learning Styles	Characteristics
Activists	People who learn best by doing and those who open themselves up to learning and new experience.
Theorists	Learners who thoroughly enjoy the learning process through models and facts.
Pragmatists	Individuals keen on the practical application of ideas during experiments.
Reflectors	Individuals who prefer learning and reflecting through observing other people's experiences.

Honey and Mumford also initiated the Learning Styles Questionnaire (LSQ) to aid learners in identifying their Learning Styles [99]. To date, two types of questionnaires exist, one with 80 questions; and the other with 40 questions. Though the LSQ is quite popular and commonly used for human resource purposes, its accuracy and validity are unknown. Furthermore, since the questionnaire is a commercial solution and not free, it's used by individual learners is very limited [100]. Thus, the Honey and Mumford model has restricted applications in the case of e-learning training designers and instructors while they identify appropriate learning styles for instruction purposes through digital media [100].

2.2.3 VARK learning style model

The VARK (an acronym for *Visual*, *Aural*, *Read/write*, and *Kinesthetic*) Learning Style model is one authored by [7]. The model focuses on representing how learners choose to perceive and absorb information and is based on four learning modes, particularly, *visual learning*, *auditory learning*, *physical learning* and *social learning*. Using these learning modes which are also shown in Table 2.3, it is up to learners on what mode to choose in order to experience a first-class learning process. At times, a single learner can demonstrate a preference for multiple Learning Styles. Such learners are known as multi-model learners. So as to identify the learning mode for a particular learner, the VARK model makes use of a questionnaire consisting of 16 questions [7].

Table 2.3: Learning modes as per the VARK model [7]

Mode	Tendency in learning process
Visual (V)	Learner prefers to learn by seeing, i.e., through visual aids such as pictures, videos, etc.
Aural (A)	The desire to learn by listening, i.e., with the help of audiotapes, lectures and music.
Read/Write (RW)	Learners inclined towards learning through reading and writing in the form of presentations.
Kinesthetic (K)	Learning through practical real encounters and hands on work instead of listening or watching.

2.2.4 Myers and Briggs Learning Type Model

Taking inspiration from Carl Jung [101] who postulated that every individual possesses one of four personality traits, Katherine Briggs and Isabel Myers developed the Myers-Briggs Type Indicator ((MBTI) which is primarily a questionnaire to classify individuals in four categories based on their personalities. These categories, popularly known as dichotomies include *Extroversion/Introversion*, *Sensing/Intuition*, *Thinking/Feeling*, and *Judging/Perceiving* [94], adding up to a total of (2^4), i.e., 16 patterns.

- The ***Extroversion/Introversion*** dichotomy is based on an individual's attitude, where extraversion refers to an extrovert's response to his/her surroundings, such as people or things, while introverts delve upon their own ideas.

- Reflecting on functions, the ***Sensing/Intuition*** category refers to an individual's perceptions to gather information. These are those who collect information through their five senses, while others are those who use intangible information by believing in their intuition.

- Referring to an individual's method of judging, the ***Thinking/Feeling*** dimension is used by individuals to arrive at a rational decision. Thinkers are those who make use of logical quantifiable frameworks to arrive at a decision while feelers are those that judge based on taking their feelings into consideration and empathising with the situation above all.

- The final dimension which includes ***Judging/Perceiving*** refers to an individual's preferences, whether they want to judge by making a structural and sequential approach or judge by being flexible and keeping options open.

According to [102, 103], MBTI has limited applicability for recommending the most suitable learning materials in e-Learning.

2.2.5 Felder and Silverman Learning Styles Model

Reflecting on popular learning style models such as that of Kolb's [1] and the work of Myers and Briggs [94], Richard Felder and Linda Silverman [2] developed a model consisting of four dimensions to provide a balance between extremes. Illustrated in Figure 2.2 are the characteristics of the model as prescribed by Felder and Silverman. Each of these dimensions is defined as below:

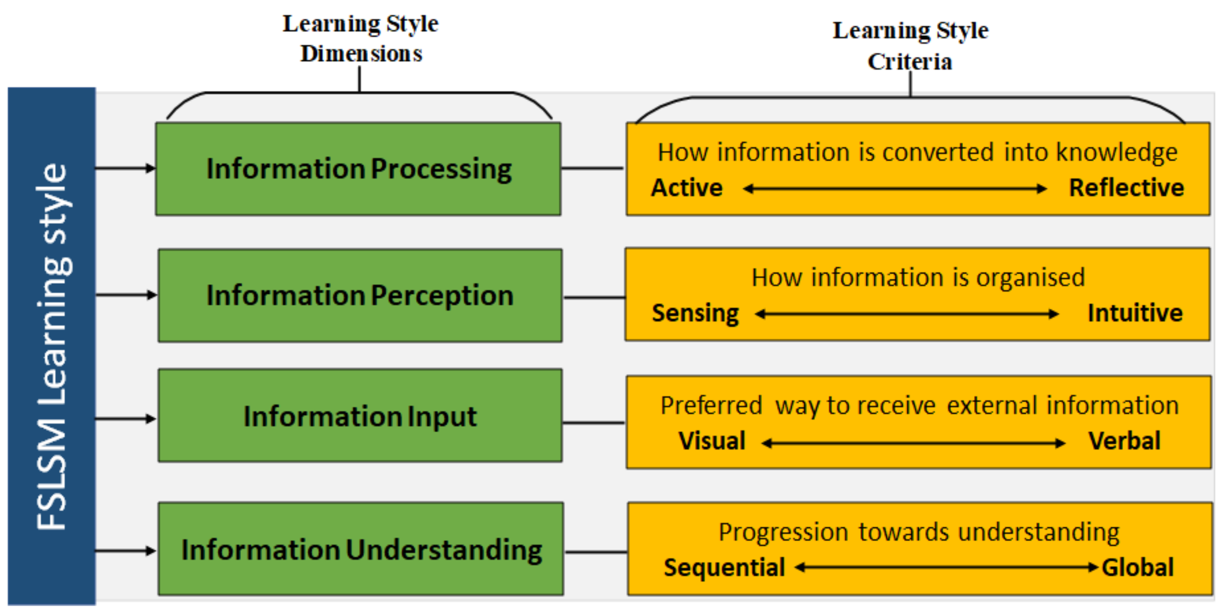


Figure 2.2: FSLSM learning style model [2]

- ***Active/Reflective*** cites the ability of a learner to be able to process information. While an active learner learns best by practising and working in groups, a reflective learner prefers learning by thinking things through and working individually.
- ***Sensing/Intuition*** deals with how individuals prefer to perceive information. Sensing learners choose to deal with concrete facts whereas an intuitive learner works based on abstract concepts and ideas.
- ***Visual/Verbal*** is related to an individual's preference on how information is presented. While visual learners process information better when presented through graphics and presentations, verbal learners perform better by listening to verbal instructions and lectures.

- ***Sequential/Global*** is concerned with a learner's ability to understand information. Generally, a sequential learner inclines towards thinking in an orderly manner, in sequential steps, a global learner prefers learning by integrating various ideas and in large leaps.

Felder and Silverman proposed a questionnaire comprising 44 questions known as the Index of Learning Styles (ILS) (see Appendix A), to aid individuals in finding out what their learning preferences were. Each dimension is represented by 11 questions, where there are two possible solutions "a" or "b" for each question. For instance, a dimension characterised by Learning Styles X/Y could have two solutions, either "a" which corresponds to preference for X, or "b" which indicates inclination towards Y.

To determine a learner's learning style within a specific dimension using Felder-Silverman's questionnaire, it is sufficient to count the number of answers "a" and the number of answers "b" on all 11 questions corresponding to that dimension followed by calculating the difference between these two numbers. Obviously, this score is an odd number between 11 (all responses of the learner are equal to "a") and -11 (all responses are equal to "b"). A learner with a score of 1 or 3 (-1 or -3) has a mild preference for X (resp. for Y); yet is essentially well balanced to learn in a teaching environment that favours X or Y. For a score of 5 or 7 (-5 or -7), the learner has a moderate preference for X (resp. for Y) and will learn more easily in a teaching environment that favours X (resp. Y). Finally, a score of 9 or 11 (-9 or -11) indicates a strong preference for X (resp. for Y).

2.2.6 Choice of learning style model

For this research the FSLSM has been employed (as defined in Section 2.2.5 above) to constitute the student profile (Learning Styles) as well as the learning object profiles, due to the below justifications:

- Firstly, as displayed in Table 2.4, in addition to integrating the work of various popular Learning Style models such as Kolb [1], Myers and Briggs [94], Honey and Mumford [6] and the VARK model [7], the FSLSM is known to be the only existing model till present which covers all categories. For instance, the *Active/Reflective* dimension in the FSLSM replicates the *Activist/Reflector* Learning Style as defined by Honey and Mumford. Furthermore, the *Sensing/Intuitive* aspect in the FSLSM is identical to the *Concrete/Abstract* mode as specified by Kolb [104, 105] in addition to the *Pragmatist/Theorist* in Honey and Mumford's model. Moreover, the *Visual/Verbal* group corresponds to the *Visual/Read-Write* mode within the VARK model. Moreover, FSLSM is different than the existing learning styles model by considering the way student prefer to understand information in terms of (sequential and global) which enhance the adaptation of e-learning system (See Sect. 2.2.5). Hence, due to its close proximity and links to various other Learning Style models, the FSLSM [2] has popularly become known as a model to be used in the development related to Learning Styles in personalised e-learning systems [10, 106–108].

Table 2.4: Learning styles criteria grouping according to learning style models

Criteria	
Active Group	**Reflective Group**
Activist (Honey and Mumford LSM)	Reflector (Honey and Mumford LSM)
Accommodating (Kolb LSM)	Diverging (Kolb LSM)
Active (Felder-Silverman LSM)	Reflective (Felder-Silverman LSM)
Sensing Group	**Intuitive Group**
Pragmatist (Honey and Mumford LSM)	Theorist (Honey and Mumford LSM)
Converging (Kolb LSM)	Assimilating (Kolb LSM)
Sensing (Felder-Silverman LSM)	Intuitive (Felder-Silverman LSM)
Kinaesthetic (VARK LSM)	
Visual Group	**Verbal Group**
Verbal (Felder-Silverman LSM)	Verbal (Felder-Silverman LSM)
Read/Write (VARK LSM)	Read/Write (VARK LSM)
	Auditory (VARK LSM)
Sequential Group	**Global Group**
Sequential (Felder-Silverman LSM)	Global (Felder-Silverman LSM)

- Another reason for choosing the FSLSM over other Learning Style models is due to the fact that the present version of the ILS questionnaire is reckoned to be valid, reliable and appropriate to aid in identifying Learning Styles. Several studies conducted in the past related to the reliability and the validity of this questionnaire set forth by Felder and Silverman have emerged out to be quite positive [109–112]. As reported by [113], the FSLSM ILS is one of the small handful of questionnaires that score acceptably well in addition to having decent standards for reliability and validity.

- Finally, being utilised quite often in technology-enhanced learning, many scholars [53, 114–116] also assert that the FSLSM is a Learning Style model most suitable for use during the implementation of adaptive learning systems, as shown in Table 2.4. For example, CS383 software [117] makes use of three dimensions of the FSLSM, in particular, the dimensions of *Global/ Sequential*, *Visual/Verbal* and the *Sensitive/Intuitive*. Similarly, both the MASPLAG system [118], and the Task-based Adaptive learNer Guidance On the Web system, TANGOW [119] utilise the FSLSM by considering two dimensions, *Sensitive / Intuitive* and *Global / Sequential*. According to [54], this approach of the FSLSM is considered to be the most appropriate methodology for use within educational systems.

After reviewing the concept of Learning Styles and relevant models, it is understood that the main goal of e-learning systems is focused on the correct identification of student Learning Styles in order to make use of the correct personalised resources and recommend appropriate instructional material, all of which are discussed in the following sections of this chapter.

2.2.7 Learning Style Detection

There are two main approaches to detect student learning styles which are widely agreed on in the literature [107, 120–123]: explicit approach (Questionnaire-based) and implicit approach (Behaviour-based) [124, 125], as follows.

2.2.7.1 Explicit approach

Following the acquisition of learners' responses to a learning style questionnaire, the characteristics and preferences of every learner is contained within the explicit modelling approach. Numerous examples of organisations known to utilise this approach include CS383 [10], INSPIRE [126] and iWeaver [16]. Despite extensive use, the explicit approach has its own set of limitations which diminish the precision of Identification of Learning Styles [127]. Firstly, this method identifies Learning Styles of student based on the results from a one-time survey, which is against other previous works which state that learner characteristics change over time depending on the situation [128–131]. Secondly, the explicit approach works under the presumption that learners are already aware of their Learning Styles. Therefore, encouraging students to be able to give out adequate explicit information proves to be a challenging endeavour [128, 132].

2.2.7.2 Implicit approach

The implicit methodology utilises a dynamic modelling perspective to deduce students' learning styles. Implicit approaches are commonly used to enhance the acquisition of Learning Styles by observing students' behaviour in addition to their interactions with the system [133]. This approach uses numerous computer algorithms and programs to spontaneously ascertain an individual's personal traits [134]. Since the implicit approach result in frequent updates to learner information [135], it is regarded to be more precise than the questionnaires [136] since it responds immediately to variations in learning behaviour patterns [26, 137]. Additional approaches which aid in the automatic identification of Learning Styles are presented in Fig 2.3 [3] as follows:

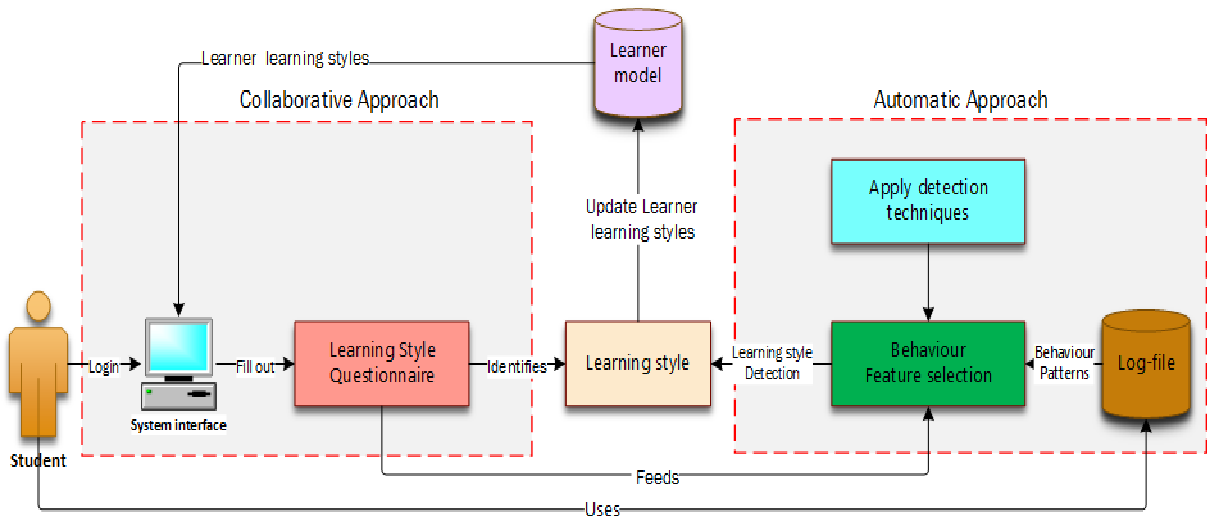


Figure 2.3: Identification of learning styles and automatic detection of learning styles [3]

- The **Data-driven approach** which is centred on constructing a model that simulates the Index of Learning Styles (ILS) questionnaire builds a profile by utilising sample data. Recurrent methods used as part of this approach include Bayesian Network, Decision Tree and Neural Network, Naive Bayesian tree algorithm, Hidden Markov and Genetic Algorithm [40, 122, 138–141].

- The **Literature-based approach** utilises student behaviour to recognise Learning Styles. A basic rules-based methodology using behaviour is carried out to calculate Learning Styles. Examples of such rules include data on the basis of the amount of matching hints and time spent on learning objects among others [52, 53, 93, 142–145].

Overall, collecting student information implicitly by using any of the previously mentioned mechanisms (data-driven or literature-based) has both advantages and disadvantages. One obvious advantage is the fact that students are not burdened with filling forms and questionnaires or rating learning objects [146]. On the other hand, one of the most important concerns about an implicit information collection is that extracting user interests implicitly requires adopting different machine-learning techniques to analyse the collected information to know students learning styles. In other words, it is quite inconvenient to translate user behaviour into user preferences, since the level of preciseness depends on the assumption that user behaviour has been interpreted accurately. Therefore, although implicit mechanisms take the burdens off users, they require more complex processes than the explicit ones. It is also important to note that implicit techniques allow for a simpler access to information by automatically updating the system once the student interacts with it.

Nevertheless, when a students' Learning Style is being detected using the implicit approach, an issue of a 'cold start' exists almost regularly [147], since (i) not much information is obtainable to build profiles [148] and (ii) the system is incompetent while inferring information for new users.

Table 2.5 displays information related to systems which utilise an explicit or implicit approach or both. In order to prevent drawbacks as well as improve the validity of Learning Styles identification, this research will take advantage of the two approaches described above (implicit and explicit) to detect students' Learning Styles. The FSLSM questionnaire [2] ascertains the first Learning Style of a student. During the course of the system usage, the student's Learning Style is dynamically adjusted in accordance with the behaviour of the user (i.e. interactions with the system) (see Chapter 4 and Chapter 5).

In an effort of achieving the intention of understanding student Learning Styles, a review and analysis of current E-learning management systems is presented in the following section.

2.3 E-learning and educational technology

This section which introduces the background associated with the concept of e-learning followed by platforms used as part of e-learning is split into two sub-sections. Section 2.3.1 introduces the various definitions of the term "e-learning" while Section 2.3.2 discusses current e-learning technologies used to deliver and manage e-learning documentation and activities, in addition to drawbacks of the current learning management system.

2.3.1 Definition of e-Learning

E-learning has widely expanded to supplement traditional classroom-based learning [157], especially in the educational sector. Usually, learners can access learning resources whenever through the e-learning technology. The term "e-learning" which comprises of two parts - 'e' which stands for electronic and 'learning' to mean "electronic learning" [158] is also known by various other names including virtual

Table 2.5: Summary of student's learning styles detection approaches in existing adaptive e-learning systems

		Student modelling approach	
System Name	**Learning Style Model**	**Explicit Modelling (Questionnaire)**	**Implicit modelling (Behaviour pattern)**
MAS-PLANG [118]	FSLSM	✓	✓
WELSA [149]	Unified LS Model		✓
TANGOW [119]	FSLSM (understanding and perception)	✓	
shaboo [150]	FSLSM	✓	
e-Teacher [18]	FSLSM (perception, processing, and understanding)	✓	✓
CS383 [10]	FSLSM	✓	
iWeaver [16]	Dunn & Dunn Mode	✓	
LSAS [19]	FSLSM (Sequential /global)	✓	
PLORS [151]	FSLSM	✓	
DeLeS [93]	FSLSM		✓
Protus [152]	FSLSM	✓	✓
OSCAR CITS [153]	FSLSM		✓
AES-CS [154]	Field dependence and field independence	✓	
LearnFit [155]	Myers-Briggs Type Indicator (MBTI)	✓	✓
AHA(adaptive educational hypermedia) [156]	Multiple learning style models	✓	
INSPIRE [126]	Learning style	✓	

learning, online learning and web-based learning [159]. It is also necessary, however, to define 'learning' before defining 'e-learning'. Generally, Learning is regarded as the process whereby learners develop new skills or modify existing knowledge and skills in order to enhance their performance [160].

In general, e-learning is formally defined as "an approach to learning and teaching, representing all or part of the educational model applied, and based on the utilisation of electronic devices and media as tools for enhancing access to training, interaction and communication promoting the adoption of new ways of understanding and developing learning" [4, 161]. Duggleby [162] defined online learning as a learning approach using technological devices including computers and handheld devices.

Moreover, the authors in [4] have categorised the various interpretations of e-learning into four key classes, as displayed in Fig. 2.4 that include the following.

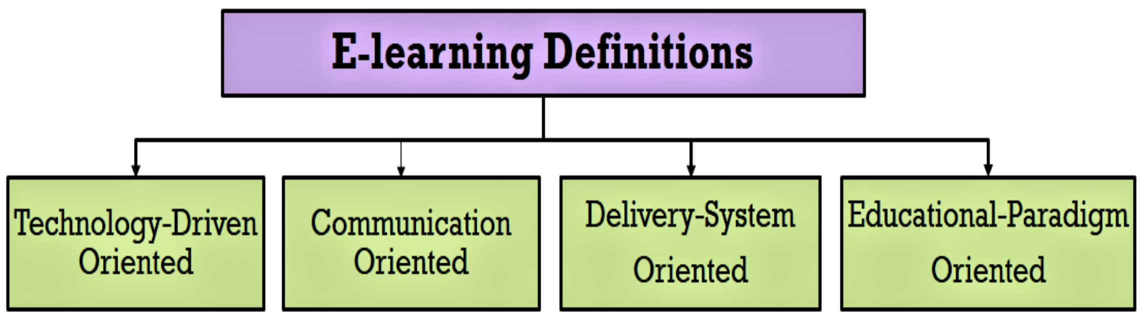


Figure 2.4: Definitions of e-learning [4]

- **Technology-Driven-Oriented Definitions:** Centred on the technical characteristics of e-learning, within this class, the concept of e-learning is defined as *"the use of technology to deliver learning and training programs"* [4].

- **Communication-Oriented Definitions:** Focused on communication tools, this class is related to e-learning where *"E-learning is learning based on information and communication technologies with pedagogical interaction between students and the content, students and the instructors or among students through the web"* [4, 163].

- **Delivery-System-Oriented Definitions:** This definition class is associated with the resources utilised for delivering e-learning. Within this class, e-learning is defined as *"the delivery of education (all activities relevant to instructing, teaching, and learning) through various electronic media"* [4, 164, 165].

- **Educational-Paradigm-Oriented Definitions:** Concerned with the educational side of e-learning, class is described as *"the use of new multimedia technologies and the Internet to improve the* *quality of learning by facilitating access to resources and services, as well as remote exchange and collaboration" [4, 166, 167].*

It should be noted that a common point among the different definitions and classes is associated with the use of technology and technological devices including computers and handheld devices in order to access and share information. For the purpose of this study, it will be useful to provide a simple definition to the concept of e-learning by considering the aforementioned classes:

“ E-learning is an approach to teaching and learning, representing all or part of the educational model applied, that is based on the use of electronic media and devices as tools for improving access to training, communication and interaction and that facilitates the adoption of new ways of understanding and developing learning ”.

This concept of e-learning is utilised in the educational sector where it is used as a resource to provide prompt support, online student management, as well as formative and summative assessment feedback to students [157]. In contrast to conventional face-to-face methods, the efficiency of e-learning is dependent on the efficient application of techniques to fulfil teaching objectives in addition to the effective coordination of student-teacher interactions [168]. Several e-learning platforms have been developed since the middle of 1990. The next section will provide an overview of existing e-learning platforms and showing out the drawbacks of these platforms.

2.3.2 E-Learning platforms

E-Learning platforms are put to use by numerous academic institutions in order to enrich learning activities by granting consistent access to e-Learning tools and applications from everywhere. Generally, the prime improvement goals of most e-Learning resources include strengthening collaboration, efficient management of learners, learning materials, notifications, assessments and results among others. Today, e-Learning platforms have come to be known by various terms such as Learning Management Systems (LMS), Course/Content Management Systems (CMS), Virtual Learning Environments (VLE) and Student Management System (SMS). These platforms can be defined as follows:

- Course/Content Management Systems (CMS) are defined as *"a collaborative learning environment containing tools for developing and delivering courses with the aid of the Internet"* [169]. While [170] revealed that the terms CMS, LMS and VLE could be applied interchangeably, [171] provided a distinct definition to VLE by terming it as *"a collection of software tools supporting academic administration, teaching and research"*, such as WeLearn [172].

- An online software used by educational institutions, known as a Student Management System (SMS) is made use of to organise students' data [173] and provide services related to students, faculty, coursework, fees, tests and results. Some examples of the SMS used in a commercial setting include PeopleSoft and Banner [173].

- Learning Management Systems (LMS) are broadly utilised to monitor numerous online educational systems for managers and are defined as *"software that automates the administration of training events"*. Through the LMS which handles multiple providers; users are registered, courses are tracked, information is recorded, and relevant reports are published and provided to the management. It is important to note that an LMS only manages courses generated by other sources and does not have potential for authoring on its own [173]. Few examples of these systems include Blackboard [174], MOODLE (Modular Object-Oriented Dynamic Learning Environment) [175], and Sakai [176].

The above-described e-learning platforms allow the provision of numerous ways of online learning in addition to making distance learning easier. At present, various higher educational institutions make use of popular virtual platforms such as Blackboard [174], MOODLE [175], and Sakai [176], that deliver coursework with several online features which learners can make utilise for effortless online learning. Figure 2.5 illustrates the most popular Learning Management Systems. The application of e-learning strategies within the educational sector and corporations is quite a customary practice in the 21st century. Yet, numerous e-learning frameworks even now pursue a traditional homogenous approach, i.e. a "one size fits all" model to deliver identical learning materials regardless of the distinct Learning Styles and preferences of learners [50] (see Sect.2.1). As noted by [52], Learning Management Systems are usually course-centric rather than learner-centric and it is due to this characteristic that LMSs are unable to offer personalised learning to learners. According to [177] as well, this issue of e-learning platforms poses as a limitation due their inability to offer learner-centric content. Hence, the concept of adaptation is a major challenge associated with the concept of e-learning, specifically in learning environments where learner preferences are diverse.

Due to the demand in moving away from standardisation, a modern trend termed as adaptive e-learning which takes into account various learner preferences has begun [52, 53]. The aim of adaptive

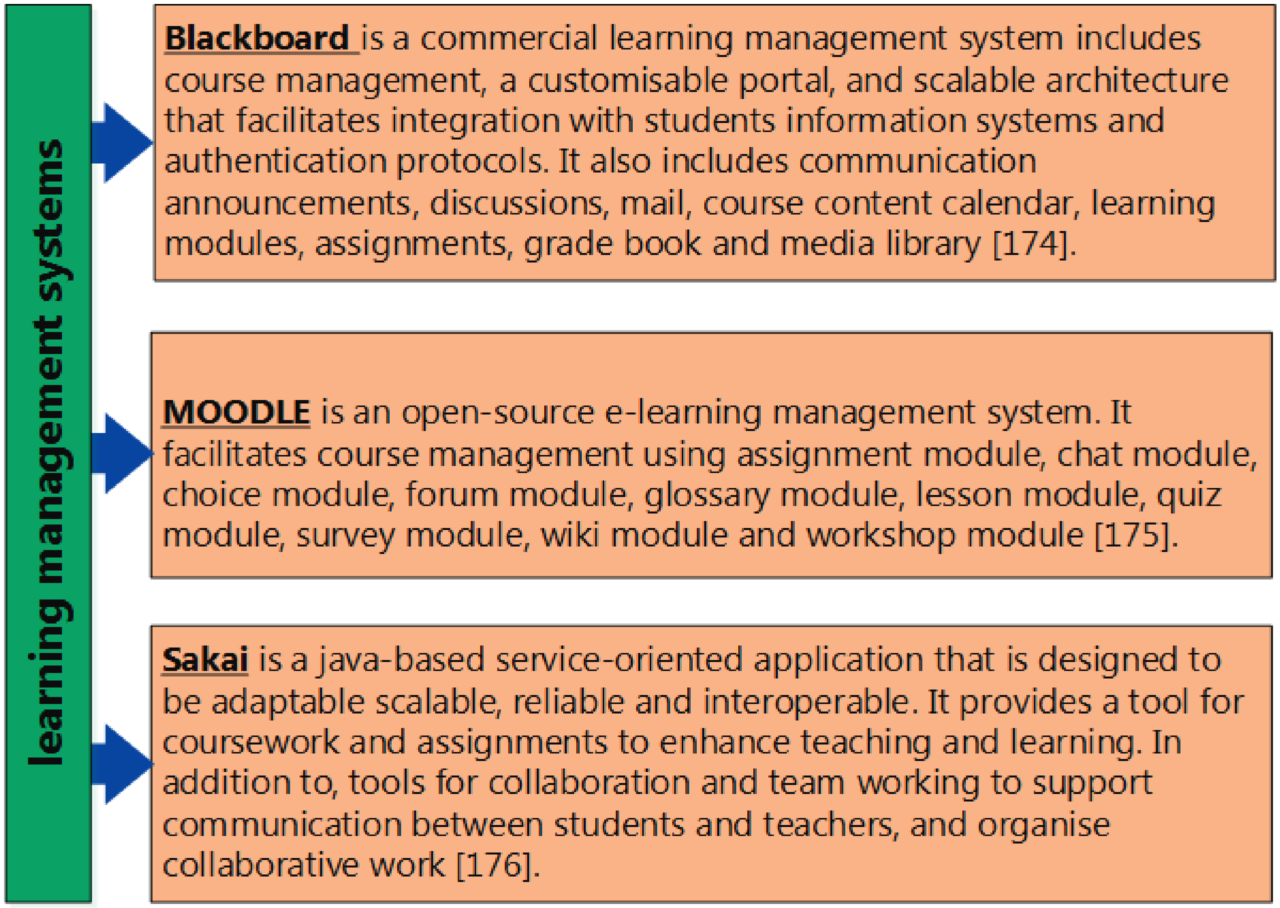


Figure 2.5: The most popular learning management systems

e-learning systems including overcoming the existing challenges of traditional e-learning practices, as shown in Table 2.6.

Table 2.6: Comparisons between the traditional and adaptive e-learning environment

Criteria	Traditional e-learning environment	Adaptive e-learning environment
Learning Style	One-to-many	One-to-one or many-to-one (i.e. One, or many learning materials for one learner).
Learning Process	Designed for an average learner	Learning content relies on the characteristics of individual learners, i.e. prior knowledge, needs, skills, learning style, and preferences.
Learning Materials	Determined by the educator	Depends on the learner's requirements.

On the basis of this analysis, the following section reviews the background related to adaptive e-learning, including the concept and components of adaptive e-learning in addition to further adaptive system related work within the field over the last years.

2.4 Adaptive e-learning

Adaptive e-learning is a specialised class of e-learning, in accordance with the philosophy that learners differ from one another in several ways. For instance, learners differ with respect to their individual knowledge levels, Learning Styles (LSs), and cognitive abilities. Thus, it is essential for e-learning platforms to adapt and deliver content depending on learner preferences [178, 179]. This is where adaptive e-learning comes in by allowing the integration of learner characteristics into learner models to deliver adaptive content. Improving learning efficiency and learner satisfaction rates are the major reasons for the significant push in the development of adaptive e-learning models [180, 181].

As stated by [182], an adaptive e-learning system constitutes *"an interactive system that personalises and adapts e-learning content, pedagogical models, and interactions between participants in the environment to meet the individual needs and preferences of users if and when they arise"*.

Interchangeable terms such as intelligent adaptive/adaptable systems, adaptive educational systems and personalised learning systems are employed in literature to describe the concept of adaptive e-learning. Nevertheless, a significant difference exists between an adaptable and an adaptive system where the former represents a user-initiated adaption technique, while the latter indicates system/automatic-initiated adaptation techniques without direct user intervention [183]. Generally, adaptation is realised in either of three forms, namely, adaptive content presentation, adaptive content sequencing as well as adaptive navigation instruments as shown in Table 2.7 [126, 184].

2.4.1 Components of adaptive e-learning systems

Adaptive e-learning environments are adopted are closely linked to well-organised and structured models. Figure 2.6 shows the major components of adaptive e-learning systems: student model, a learning objects model, and an adaptive engine; all of which are described in the following sections.

Table 2.7: Adaptation categories

Adaptation categories	Description
Adaptive presentation of academic content	Content adapted to tailor to learner preferences constitutes this approach. In addition to ensuring adaptive content to learner characteristic, another aim of this category involves a substantial increase in the speed as well as quality of learning [126, 185]. Examples of this category include Arthur [186] and Computer Systems hypermedia courseware (CS383) [117].
Adaptive content sequencing	The purpose of this category involves the order in which the content is presented and delivered to suit various learner preferences, such that the presentation sequence has a significant effect on the learning process [184, 187]. Examples of this category include Adaptive Courseware Environment (ACE) [188] and the (INtelligent System for Personalised Instruction in a Remote Environment (INSPIRE) [126].
Adaptive navigation tools	Cites the provision of proper orientation of e-learning tools in order to enhance user experience based on students' habit and preferences by the adjustment of visible links that control learning orientation such as that of Adaptive Educational System based on Cognitive Styles AES-CS [154].

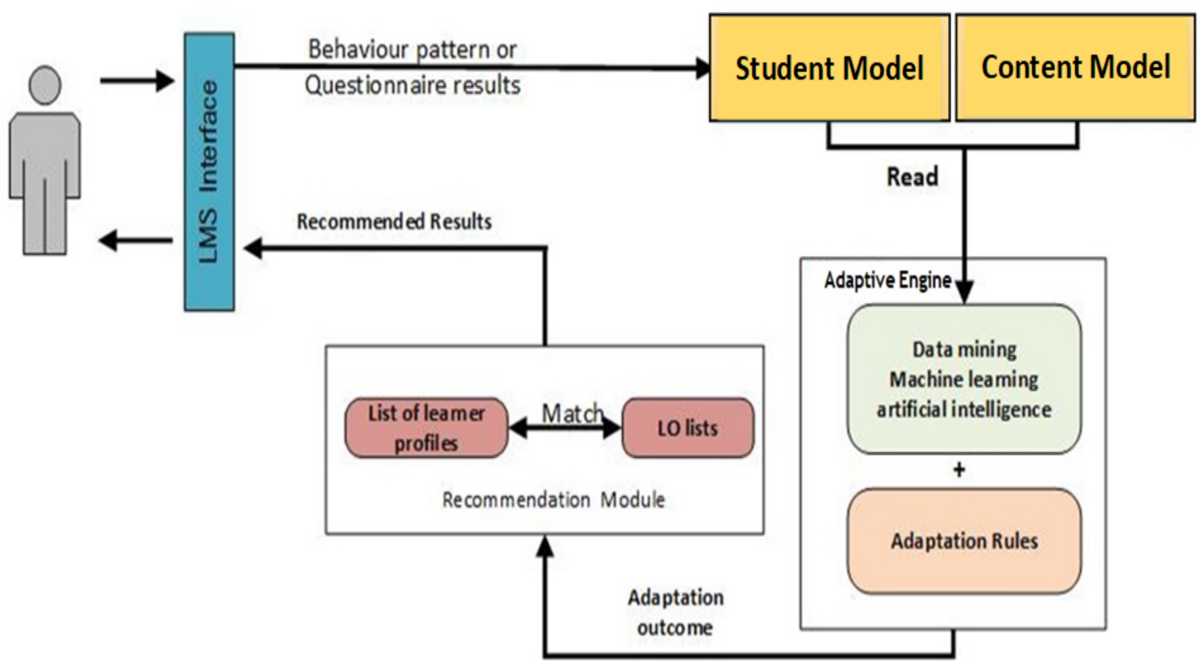


Figure 2.6: Adaptive e-learning systems' components

2.4.1.1 Student Model

A student model is responsible for tracking an individual learner's data in order to adapt itself to the learner's preferences [189]. In this context, [190] expresses that a student model is considered to be a crucial piece of individualised behaviour in adaptive e-learning systems and strongly depends on the way in which knowledge about a student is modelled internally. This process of building and updating a Student Profile (SP) is known as student modelling, whose phases are described below in Figure 2.7. Furthermore, student modelling can be classified [5] into static or dynamic modelling as described below:

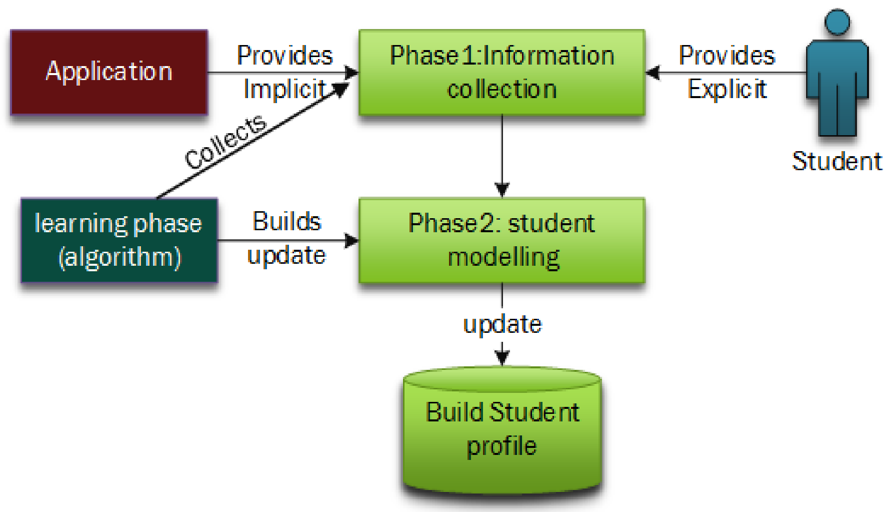


Figure 2.7: Student modelling in e-learning systems [5]

- ***Static modelling*** describes the process of initialising student information once, generally at the time of student registration [191].

- ***Dynamic modelling*** is associated with the process of updating student models consistently and respond to changes during the course of enrolment [191].

2.4.1.2 Content Model

Content model generally comprise information related to the knowledge domain of course content in order to facilitate an adaptive course delivery. The concept of Learning Objects (LO) is made use of to reduce unnecessary time and effort taken up to develop educational material. [192] explains Learning Objects to be organised digital materials of learning used in various learning environments and annotated using metadata for the purpose of describing and manipulating them. As defined by the IEEE [193]. Most broadly used standards related to metadata in e-learning include those of the Dublin Core [194] and IEEE LOM [193].

2.4.1.3 Adaptive Engine

The adaptive engine is an algorithm that integrates information from the student and course models to select appropriate course learning objects to present to the student. There are two essential types of adaptive engines used in for adaptive learning systems [195, 196]:

- ***Rule-based*** using conditional 'if-then' logical decisions. This approach builds its system as a branching architecture. This approach is clear and simple; however, it becomes more and more difficult to process the branches as the number of components of the domain model increases [196–198].

- ***Algorithm-based*** using mathematical functions to analyse students' behaviour patterns. This approach is considerably more complex than the Rule Based technique. Algorithm based selection often uses machine learning techniques to learn more about the contents and the students' behaviour. These techniques employ highly complex algorithms for predicting which choice of content is likely to be most successful in terms of learning outcomes [199].

Finally, the recommendation module executes the adaptive matching rules coming from the adaptive engine and provides the recommendations, as explained in Section 2.5.1.

2.4.2 Adaptive e-learning evaluation methods

In general, the evaluation of adaptive e-learning systems that deploys students' profiles is considered to be complicated in addition to being expensive [200]. There is no standard way to evaluate adaptive systems, but each system might propose different strategies which are tailored based on their purpose and structure. On the basis of the literature, three main approaches commonly used to evaluate recommendation systems include offline, user-centred studies and online evaluations [27].

- Existing datasets have been extensively used in **offline evaluations** to measure the performance of recommender systems through statistical analysis. As will be explained in the next chapters, we have used students' datasets from the AAST-MOODLE log-file so as to independently test the performance of each module of ULEARN system. Within this type of evaluation, there is no need to deploy the system or interact with real users in an online environment. Another advantage includes the ability to reuse datasets for testing different algorithms and parameters for the purpose of finding optimum performance for a system. The main drawback of this evaluation comprises the way in which it cannot measure users' opinions of the personalisation experience. However, user-centred and online approaches are proposed to address some of these limitations.

- Real students are engaged in **user-centred evaluations** for collating realistic behaviours in addition to testing the proposed system's performance [27] and comprises three steps: (i) recruiting students; (ii) creating a set of tasks to be completed by the users; and (iii) collecting and analysing student interactions and behaviours. These evaluations are also known to be a better indicator of the proposed system's effectiveness, accuracy and efficiency instead of using datasets since systems are tested in realistic settings. Many studies [201,202,202] confirm that the user-centred evaluation is suitable for evaluation of the adaptive e-learning system's accuracy and overall performance.

- A system is made use of in **online evaluations** to be evaluated within a real environment. This approach is presumed to be the best approach for evaluation purposes since thorough testing of the system can take place corresponding to real users' behaviours within a real environment [203]. Therefore, the results obtained through means of an online evaluation would have stronger evidence of the system's performance. Although online evaluations might be more reliable to assess a system's performance, they possess few challenges such as being time-consuming and complicated.

2.5 Recommender Systems

Recommender systems (RSs) comprise software applications for the filtration of information according to an individual's personalised preferences. In other words, while they are utilised in a variety of areas, these systems offer personalised suggestions to users about products which include movies and books among others [204,205]. Hence, recommender systems are said to assist in managing the problem of information overload by offering users only relevant items from a substantial number of items [27,206,207]. RSs have been widely used in the last few years in different fields including e-commerce [9,32,33], e-learning [208, 209], news [34–36], music [37], movies [210] and documents [211]. To provide recommendations, RS try to predict relevant products and offer recommendations to users based on their individual preferences [211]. Recommender systems generally use machine learning techniques to process stored data about user's interests, represented as user profiles. A ranked list of all resources available for recommendation is subsequently computed based on an algorithm that has been trained using the user profiles. In building recommender systems, historic data about users are generally stored in a user-item matrix also known as the rating matrix.

Data related to the matrix is gathered through relevant feedback which could either be explicit (ratings, votes) or implicit (clicks, purchases) [212, 213]. Table 2.8 involves five users (denoted as rows) and five items (denoted as columns). Each cell value represents user-item interactions (e.g. a rating ranging from 1 to 5), where a rating of 1 represents extreme dislike, while a 5 denotes extreme liking for a particular item. The empty cells denote the non-observed data (missing values). The task of predicting absent

Table 2.8: User-item rating matrix

	$Item_1$	$Item_2$	$Item_3$	$Item_4$	$Item_5$
$User_1$	**1**			**3**	
$User_2$		**5**	**3**		**4**
$User_3$		**3**	**2**	**1**	**3**
$User_4$			**1**		
$User_5$		**3**		**4**	

values within the matrix constitutes the aim of the RS algorithm. These missing values exist in areas where there is no data related to the users' preference. Furthermore, data within the matrix could sometimes be quite scarce in the case where the number of products a user interacts with; or the number of user ratings are few, in comparison to the total number of items [214]. Numerous approaches exist to predict user-based ratings such as Content-Based Filtering, Collaborative Filtering and the Hybrid Approach, all of which are discussed in the following section.

2.5.1 Recommendation Techniques

In the interest of predicting user preferences and ratings, numerous recommendation systems exist as mentioned in the previous section, some of which are illustrated in Figure 2.8. In this section, relevant recommendation techniques such as Content-Based Filtering (CBF), Collaborative Filtering (CF), Knowledge-Based (KB), and Hybrid Filtering (HF), shall be discussed.

2.5.1.1 Content-Based Filtering (CBF)

This recommendation approach takes into account the user's preferences and suggests products in accordance with items that the user has liked before [215–217]. The algorithm of this approach is structured

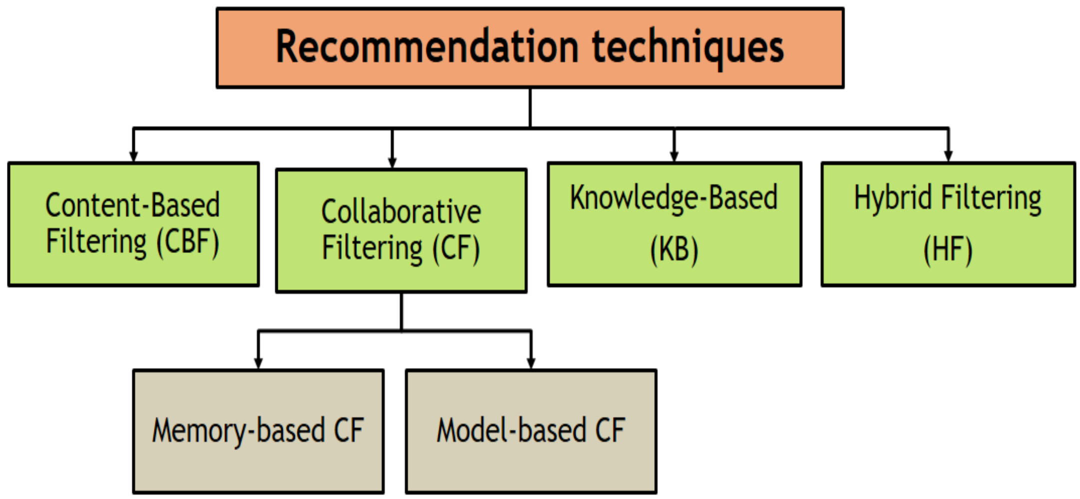


Figure 2.8: Breakdown of recommendation algorithms used for recommender systems

to perform an analysis on items in order to identify any which might be of the user's interest, as shown in Figure 2.9. An example of Content-Based filtering includes a situation wherein if the user has liked

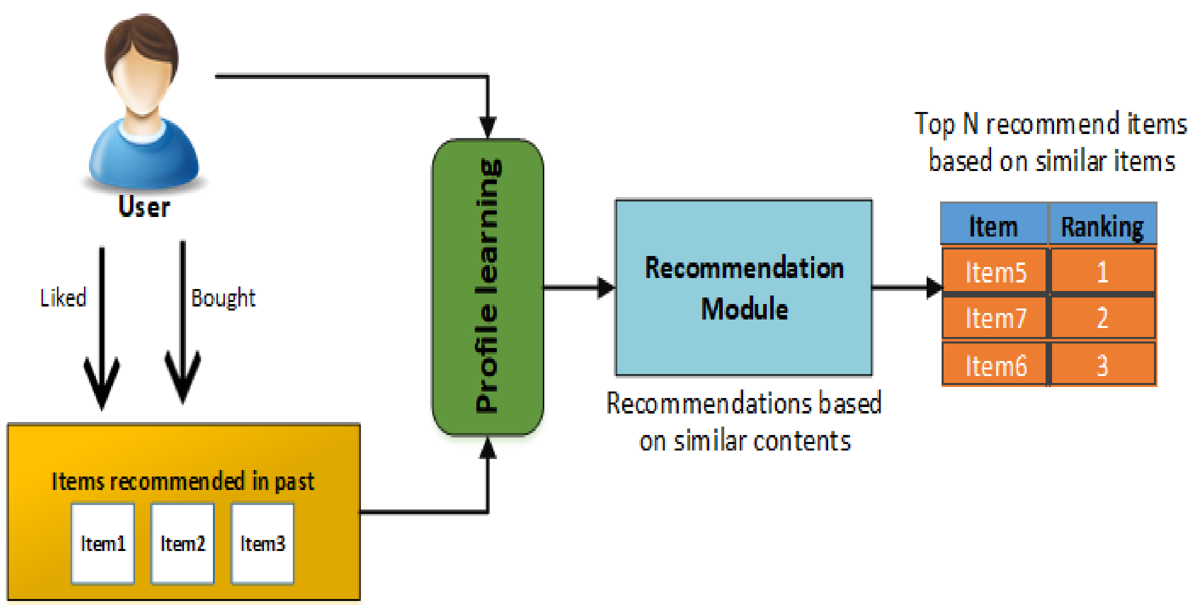


Figure 2.9: Content-based filtering recommendation based on similar items

a book related to Artificial Intelligence (AI), the system suggests similar books to the user. Four prime steps form part of this algorithm, as described below:

1. First, item information is gathered by the system; for instance, a movie recommender system would collect data related to movie titles, genres, actors, producers, etc.

2. Next, users are requested to rate items either using a binary scale (likes/dislikes) or a numeric scale (a scale of 1 to 5).

3. Based on the collected information, a user profile is created using the provided ratings. Various methods of retrieval are used for this purpose. As more information is gathered, user profiles are updated.

4. The recommendation system then matches content related to unrated items with an active user profile and subsequently assigns a score to the items.

For the purpose of deducing the similarity between an item and a user, numerous similarity approaches are applied. One such commonly used measure is that of the Cosine Similarity (see Section 2.7).

Moreover, various techniques, such as classification [216, 218], regression and clustering [219] are utilised for Content-Based recommendations due to their ability to adopt semantic item content and recommend items to users according to their preferences.

In this way, a CBF recommender system would possess the ability to recommend new as well as unpopular items to users. In addition, these systems could provide a clarification of recommended items by listing content-features on the basis of which item is to be recommended and do not experience issues with sparseness since information related to the preferences of other users is not required. However, a limitation of the CBF approach is an issue involving new users where the system is not capable of providing accurate suggestions to new users who have rated just a few items. Additionally, this approach also suffers from an over-specialisation problem where recommendations can be made only in line with preferred items within the user's profile and not those outside the scope of the user's profile. Moreover, it is not possible to distinguish between items that portray an identical group of features through this approach. Occasionally, it is not possible on behalf of the recommender system to suggest items that are very identical to users, for instance, news articles which report the same event. One more drawback of this approach is the issue related to item content dependency where it is difficult to recommend items such as images and movies that cannot be described as keywords.

2.5.1.2 Collaborative Filtering (CF)

As shown in Fig. 2.10, collaborative recommendation approaches make recommendations taking into account the user's behaviour, in contrast to CBF techniques which suggest items in line with similar content. As proposed and defined by [220], the CF technique which is a highly researched approach which uses a matrix of user-item ratings as input based on implicit or explicit interactions. Owing to many advantages, this approach is also widely used by massive online businesses such as the online retailer Amazon [221] and film rental service provider Netflix [222] where new items are released based on a review of users' behaviours to catalogue items in terms of purchasing activities and shared opinions. The

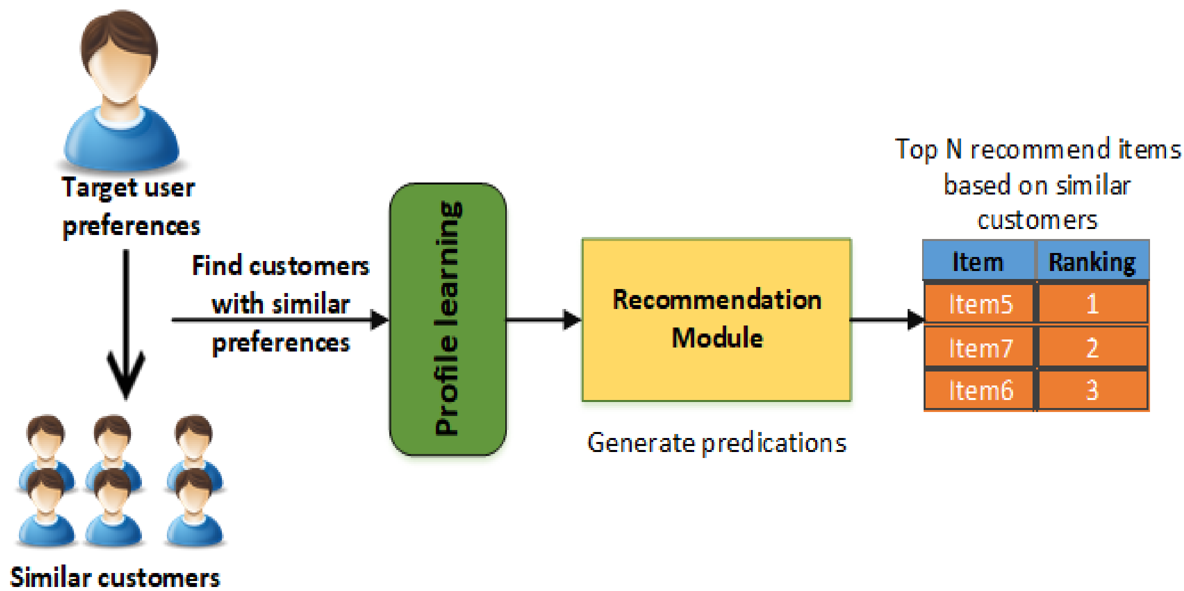


Figure 2.10: Generate predictions using collaborative filtering recommendation based on similar customers

Collaborative Filtering approach is generally classified into two categories, memory-based and model-based [29, 223], both of which are defined below:

- **Memory-based CF:** Also termed as neighbourhood-based algorithms, memory-based algorithms [224,225] are one of the most initially used CF approaches [226] which take into account the neighbourhood while predicting the ratings of user-item combinations [227]. These neighbourhoods are defined in either of the below two ways:

  - **User-based collaborative filtering:** These algorithms provide recommendations of products that were liked by similar users [29]. Therefore, the objective is to recommend ratings for items which have not yet been rated by a user. This is achieved by computing weighted averages of the ratings by users sharing the same interests as the target user [227]. To exemplify, whenever Maria and Alex have similar ratings among rated movies, a user-based CF algorithm can use the rating given by Maria to the movie Aladdin to predict the rating that Alex would have given if he had to rate it himself.

- **Item-based collaborative filtering:** These approaches provide recommendations of products similar to those that the user liked in the past [29]. Consequently, to predict an item rating given by a user, the first step involves determination of a set of with the highest similarity to the target item. The item ratings in the set are then applied to predict if the user will like the item. Hence, if Alice gave positive ratings to classic movies like Aladdin, these ratings would then be used to forecast her ratings of other animation movies, like Shrek.

The decision on which approach to use usually relies on the ratio of the number of users to the number of items. For instance, where the number of users is greater than the number of items, item-based approaches are more appropriate because they generate precise recommendations in addition to being computationally efficient [228], while user-based approaches generally produce more original recommendations [229]. These neighbourhood-based methods can be interpreted as generalisations of k-nearest neighbours' classifiers. Therefore, while these methods are considered instance-based learning methods, which are very specific in terms of the instance being predicted [230], they are usually known to not work very well with sparse rating matrices [226].

- **Model-based CF:** Such algorithms utilise machine learning and data mining to generate predictive models that will be used to predict missing entries within ratings matrix R [230]. As with supervised or unsupervised machine learning methods, these predictive models are created before the prediction phase. Examples of traditional machine-learning methods that can be generalised to the model-based CF scenario include decision trees, rule-based methods, Bayes classifiers, regression models, support vector machines, and neural networks [231].

  Two prime issues in recommender systems include accuracy and scalability. While an accurate system denotes the high ability of a system to make precise predictions of how users rate items, scalability represents the characteristic of a system to cope with a huge number of users and items. It is generally known that while memory-based techniques constitute better accuracy, model-based approaches have higher scalability [29, 231, 232]. Many applications make use of the advantages of both approaches by integrating them to form a highly efficient hybrid filtering system while curbing the drawbacks of CF approaches such as sparsity [233].

The ability of being able to work with any item types without the need for extracting item-related features constitutes a major advantage of the CF approach. Through this technique, recommendations related to new items based on other users with similar preferences are made in addition to suggesting similar items pertaining to an individual user's preferences. However, the CF approach does have its own set of limitations, some of which are related to issues of data sparseness and cold-start [213, 234, 235]. Data sparseness usually takes place when there is a substantial increase in the number of available items in addition to insufficient ratings within the rating matrix to produce precise predictions. Once the ratings acquired are few compared to the number of ratings that need to be predicted, a recommender system becomes unable to locate similar neighbours, hence producing recommendations below par.

With regards to Cold-Start (CS), this issue generally constitutes problems pertaining to the user and the item. The CS user problem, known as the new user problem, has an effect on users with a smaller number of ratings or none at all. In cases where the amount of rated items for the CS user is too few, a collaborative recommender technique cannot accurately find user neighbours by making use of rating similarity, hence failing to produce precise recommendations. Besides, the CS item problem, called the new item problem, impacts items with few ratings or none. With few ratings for CS items, the application of collaborative filtering systems fail to correctly locate similar neighbours through rating similarities and thus would be unlikely to recommend them [234].

2.5.1.3 Knowledge-Based (KB)

The concept of Knowledge-Based (KB) recommendation techniques make suggestions to users by taking into account knowledge of the users, user preferences, item information as well as recommendation criteria [236]. Generally, these recommendation systems which are used in situations where CBF and CF do not apply, retain a substantial volume of knowledge base which defines the way in which an item meets a specific user's needs. These knowledge explicit knowledge bases are generated based on presumptions between the user and the relevant recommendation [237]. One such familiar application of knowledge-based recommendation systems includes Case-Based reasoning in which Case-Based systems depict items as cases and produce suggestions through the restoration of the most identical cases to the user's query or profile [238]. As a formal knowledge characterisation methodology, Ontology constitutes the domain concepts as well as their relationships and is utilised in recommender systems to express domain knowledge [239]. As a matter of fact, there is a possibility to determine the semantic similarity between items through means of domain ontology.

Ontology-Based recommender systems [211, 240, 241] utilise Ontologies to define profiles of users and items. It is these systems that produce relevant recommendations after assessing similarities between the instances related to users and all other instances within the system's knowledge bases. Different heuristics are used for assigning the weights to super- and sub-instances. The depiction of Ontology can take several forms in various contexts. For example, it represents a model among other languages when using Web Ontology Language (OWL); while depicting an XML (eXtendable Markup Language) schema with respect to a database [242, 243].

As shown in Table 2.9, Ontology-based systems are made use of in e-Learning so as to surpass issues related to a cold-start and bring in a personalisation angle. For instance, [244] proposed a system to perform the task of recommending learning activities where ontology could be used to portray the learner profile, domain model, and learning activities. On the contrary, [245] outlined a framework to cumulate and classify learning objects, following a generation of semantic recommendations based on ontology. As mentioned above, knowledge-based recommendation techniques specifically ontology-based are applied in situations where systems based on the concepts of CBF and CF cannot be used. This is due to the advantage of KB systems not experiencing issues related to cold-start and sparsity [246]. In addition to possessing numerous benefits such as reusability, sharing domain knowledge and providing personalisation [246], reasoning the ontology aids in moving towards certain useful inferences in the recommendation process. Nonetheless, the concept of ontology does include some limitations. With the capability of being reused by anyone on the Internet, names prescribed to various classes, properties, and individuals within an ontology model prove to be a challenge. An additional such drawback which occurs during the process of designing ontologies includes the inaccurate usage of classes and individuals which call for a means of knowledge engineering through the course of ontology designing and maintenance [246, 247].

2.5.1.4 Hybrid Filtering (HF)

Due to numerous drawbacks which are difficult to overcome within the boundaries of an individual recommendation approach, several researchers have attempted to combine various methods to prevent limitations and gain the benefits of both [248]. This approach makes use of two or more recommender methods to provide the user with recommendations [228]. Merging two or more methods is usually made in an effort to overcome the limitations of one solution by incorporating another method [249]. According to [237], there consist two fundamental hybridisation mechanisms of combinations to build hybrids in recommender systems, as follows:

1. **Combining recommendation systems approach:** An approach towards building hybrid systems is to first individually implement Content-Based and Collaborative approaches, followed by com-

bining their output (the ratings) into one recommendation [250, 251].

2. **Adding Content-Based properties to Collaborative methods:** These methods utilise Content-Based profiles of users to compute the similarity between the users. These similarities are then combined with the unrated items to make the final recommendation [252, 253].

A widespread approach in existing hybrid recommendation methods includes combining CF recommendation techniques with various practices for preventing issues related to cold-start, sparseness or scalability [29, 45, 218, 251, 254, 255]. A few techniques which have been made use of are described in Table 2.9.

2.5.1.5 Recommendation techniques in e-learning

Table 2.9 summarises recently published researches conducted in the area related to adaptive e-learning systems, such as Content-Based Filtering, Collaborative Filtering, Knowledge-Based and Hybrid recommendation techniques including information related to personalisation services. Traditional recommendation algorithms still face several problems in current e-learning systems, as mentioned previously. Hence it is necessary to overcome these issues while implementing effective e-learning recommendation techniques. During the course of this thesis, a novel Enhanced Collaboration Filtering (ECF) algorithm and an Enhanced Content-Based Filtering (ECBF) algorithm will proposed, which integrate multidimensional attributes of learning object (LO), Learner's profile (Learning Styles) and learner's rating information to resolve the cold-start problem as well as the data sparsity problem. A clustering technique in particular K-means is used to group students with similar criteria preference and group learning objects according to their profiles for a better-performed recommendation. Details are presented in Chapter 5.

In the next section, a summary of clustering approaches including that of the k-means clustering algorithm will be described.

2.6 Clustering in recommender systems

Clustering is known as unsupervised learning, the concept of clustering has been used for decades in many fields, such as medical image analysis [268, 269], clustering gene expression data [270, 271], investigating and analysing air pollution data [272, 273], power consumption analysis [274, 275], and many more fields of study. Research in the field of clustering algorithms is quite vast and ranges from partitioning-based to

Table 2.9: Summary of recommendation techniques in e-learning systems

Study	Cold-Start	Issues	Used Recommendation Techniques			
			CF	CBF	KB	HF
Predict relevant learning materials to every learner based on collaboration with other learners [42].	✓	No past preferences	✓			
Recommend the most suitable learning materials for each learner based on the rating similarity with other learners [256].	✓	No past preferences or No high rated LO in past preferences.	✓			
Finding learning objects suitable for learners' preferences (knowledge level and learning style) [257].	✓	No past preferences		✓		
Recommend course learning objects based on neighbourhood rating [258].	✓	No high rated LO in past preferences	✓			
Constructed a course ontology and retrieved the course according to learner's learning styles [259].			✓		✓ (+ontology)	✓
Recommend learning materials [260].			✓	✓	✓ (+ontology)	✓
Proposed a hybrid recommender system to recommend learning items in users learning processes [261].	✓	No past preferences or No high rated LO in past preferences.	✓			✓ (+SPM algorithm)
Proposed a recommender system to store and share research papers and glossary terms among university students and industry practitioners [262].	✓	No past preferences	✓			
Recommend learning contents to users based on similarity between user profiles [263].	✓	No past preferences	✓			
Course recommendation system based on ontology and context aware e-learning [264].				✓	✓ (+ontology)	
Framework for recommending learning resources based on the learner's recent navigation history and by comparing similarities and differences among different learner's preferences and instructional content available in the e-learning system [265] .	✓	No past preferences or No high rated LO in past preferences.	✓	✓		
Recommend learning Materials based on multidimensional attributes [266].	✓	No high rated LO in past preferences	✓	✓		
Proposed approach for selecting and sequencing the most appropriate learning objects [267].	✓	No high rated LO in past preferences	✓			✓(+association pattern analysis)

hierarchical-based methods. Hierarchical methods found in algorithms such as single link, complete link and average link, are utilised to generate a nested succession of partitions in two modes: the agglomerative mode which initiates with a distinct cluster pattern and subsequently merges nearly all identical cluster pairs resulting in the formation of a cluster hierarchy; or the divisive mode which is initiated with a single cluster pattern and is followed by splitting every cluster into smaller clusters till the stopping criteria is encountered. Partitional methods made use of in algorithms including k-means and expectation maximisation methods detect clusters all at once as a partition and do not inflict any clustering hierarchy.

Clustering algorithms are normally developed based on proximity, i.e. similarity or distance. The process of clustering can be described as grouping a selection of objects into clusters with respect to dissimilarities between them. It is important to note that data objects in one cluster are similar to each other and dissimilar from objects in different clusters.

Several recommendation methods related to partition-based clustering algorithms have been proposed, as shown in Table 2.10. In this thesis, the k-means clustering algorithm has been utilised to enhance computational efficiency specifically in terms of the quality and preciseness of recommendations.

As discussed above, the k-means [280] clustering algorithm is based on partition clustering methods and has seen several applications specifically within the field of personalised e-learning [280], the AHA! e-learning system being an illustration of such an application [281]. The AHA! e-learning system develops clusters corresponding to an individual student's model and cumulates relevant data in an XML file, where the centroid of each cluster is collected. In other words, a typical user is represented by a centroid within a cluster.

The k-means approach has found various applications, some of which are described in this section. [282] utilised k-means to examine students' behaviours based on an annotation dataset of 40 students. The k-means algorithm found another application related to students when [283] administered a study to comprehend the influence of human characteristics on students' performance while listening to music. Furthermore, [284] applied the k-means algorithm to aid in building the neighbourhood. In this case, the neighbourhood was not restricted to the cluster of the user, but the user's distance to various cluster centres was used as a pre-selection step for neighbours. Successful results of this effortless and efficient technology related to clustering were achieved and it was interpreted that the results surpassed that of KNN-based CF to a higher standard. The k-means clustering algorithm is one of the simplest and most common partitioning methods [285, 286].

Table 2.10: An overview of recent researches on adaptive e-learning recommender systems using clustering algorithms

	Recommendation Techniques				
Study	CBF	CF	HF	Method	Result
[276]	✓	✓	✓	Classification & association rule (Apriori algorithm) & clustering (K-means)	K-means clustering, ADTree classification & Apriori association rule algorithm is the best combination compared with ADTree & Apriori Rule.
[48]		✓		Clustering (K-means) & association rule (Apriori algorithm)	K-means clustering and Apriori association have the best results and improve the accuracy of the course recommendation.
[277]			✓	Clustering (K-means) & Knowledge-based (ontology)	Improve the LO's recommendation taking the advantage of K-means clustering according to the student's prior knowledge.
[278]			✓	Clustering (K-means) , cosine similarity metrics & Pearson correlation	Improving in learning object recommender accuracy.
[279]		✓		K-means Bi-Directional based frequent closed sequence mining	The result shows higher LOs recommendation accurately according to the real-time up dated contextual information.
[152]			✓	Clustering Machine learning	Learners in the experimental group complete a course in less time than learners in the control group.

Standard k-clustering methods which assign objects to the closest cluster are initialised by randomly choosing k cluster centres, where k is the cluster number. Next, distance methods such as Euclidean distance (Section 2.7.1) are used to determine the similarity between the data objects and cluster centres. Each object is then assigned to one of the cluster groups with the closest centre followed by redefining the cluster centres by finding the mean vector of all objects belonging to each cluster group. Finally, the objects are reassigned according to their distance to these new cluster centres. This iterative process repeats until there are no changes in the assignment of objects to cluster groups. The traditional method aims to minimise the Euclidean target, and hence makes use of the "least sum of squares" approach. However, by making use of other distance functions excluding the Euclidean function, the algorithm could be prevented from converging. The algorithm is implemented in the following steps.

Step 1: Select K random points from the dataset as initial cluster centroids.

Step 2: Create K clusters by associating each data point with their closest cluster centroid, making use of the Euclidean distance defined Eq. (2.1).

Step 3: Recalculate the centroid of every cluster as the mean of all data points in that cluster.

Step 4: Repeat steps 2 and 3 till there is no modification in the centroids.

Enhancing the efficiency of recommender systems, clustering offers an offline feature to partition the dataset in a way that similar data is placed within the same cluster while dissimilar data is placed in various clusters, followed by which the algorithm is applied only with the highest similarity to the user. Some benefits of this algorithm include its simplicity as well as its time complexity (can be used to cluster large data sets).

The next section presents the similarity metrics often utilised in e-learning recommender systems

2.7 Similarity metrics

Considered to be the backbone of Content-Based Filtering, Collaboration Filtering and Knowledge-Based recommendation systems; similarity metrics aid in the prediction of ratings of items which have not been rated, by deducing the closeness between two featured vectors. This section explains the most widely used similarity metrics utilised for purposes related to e-learning recommendation techniques, as displayed in Table 5.3.

Let $x = (x_1, \cdots, x_n)$ and $y = (y_1, \cdots, y_n)$ be two vectors in an n-dimensional real vector space, for some integer $n > 0$.

Table 2.11: A summary of widely used similarity metrics in approaches related to e-learning recommendation systems

Study	Adaptation	Euclidean	Pearson	Jaccard	Cosine	Manhattan
[287]	Adaption of learning objects.				✓	
[288]	Calculating the degree of similarity between students.		✓		✓	
[289]	Measurement of similarity between learners' rating vectors.		✓			
[151]	Compute similarity between learners.	✓				
[290]	Calculate the similarity among students.					✓
[291]	Learning objects self-organisation process using (cosine similarity) and matching the similarity degree between learners and LOs using (Jaccard coefficient).			✓	✓	
[260]	Calculate the similarity between the user profile and the course profile				✓	
[292]	Measuring the similarity between users to recommend courses.				✓	
[293]	Calculate user similarity for the suggestion of the course.				✓	
[294]	Recommend individual learners' elective courses based on comparison of single course template with that of course templates in the same curriculum to deduce a similarity between learners.		✓			
[295]	Calculate similarity with other learners (rating and learning styles similarities) to recommend learning materials.		✓		✓	

2.7.1 Euclidean Distance

The Euclidean Distance which is always greater than or equal to zero is the most widely used measure to compute the distance of two vectors x and y as defined in by Eq. (2.1) [296]:

$$D(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.1}$$

In the case where the vectors are identical, the result would be equal to a value of 0. Furthermore, when the coordinates of both vectors range from 0 to 1 (which is the case for the feature vectors used in this study to represent the student Learning Styles and the learning object profiles, see chapter 3), i.e. $0 \leq x_i, y_i \leq 1$ for all $1 \leq i \leq n$, then the Euclidean distance can be normalised as follows:

$$d(x,y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.2}$$

Note that $0 \leq d(x,y) \leq 1$ and the greater the value of $d(x,y)$ the more dissimilar x and y are. Hence, the derivation of Eq. (2.2) will result in a similarity matrix as defined below.

$$Sim_d(x,y) = 1 - d(x,y) \tag{2.3}$$

Though the Euclidean Distance is widely used in algorithms such as K-means and Fuzzy C-means [297] for the purpose of clustering continuous data, it possesses its own limitations, i.e., (i) when two data vectors possess no common attribute values, they might have a smaller distance than the other pair of data vectors which contain identical attribute values [298, 299]; and (ii) in the case of Euclidean distances, the largest-scaled feature would be dominant over others, hence, the normalisation of continuous features could be a solution to this issue [298].

2.7.2 Pearson Correlation

A measure of the linear dependence between two vectors x and y, the Pearson correlation is described as the ratio of the covariance of the two vectors by the product of their standard deviations (which acts as a normalisation factor) [300], as shown in Eq. (2.4).

$$P(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.4}$$

Where $\bar{x}$ and $\bar{y}$ are the mean values of x and y, respectively.

Usually, the coefficient $P(x,y)$ varies between -1 to 1, where the value 1 represents perfect negative linear dependence, 0 represents no linear dependence, and 1 represents perfect positive linear dependence. It is important to note that while positive values demonstrate similarity with 1 depicting perfect similarity, negative figures represent dissimilarity. Despite being sensitive to outliers, the Person measure is invariant to linear transformations [298, 301].

2.7.3 Cosine Similarity

A widely used similarity metric, the cosine similarity computes the ratio of the scalar product by the product of magnitudes by measuring the angle between two vectors [302], as computed in Eq. (2.5).

$$c(x,y) = \frac{x.y}{||x||.||y||} \tag{2.5}$$

Generally, the values of $c(x,y)$ range from -1 to 1, and from 0 to 1 if the x and y coordinates are non-negative values. Effectively, this thesis is concerned with values between 0 and 1 where the value 0 represents no similarity and 1 constitutes perfect similarity. The Cosine correlation is invariant to rotation but is variant to linear transformations in addition to being independent of vector length [298].

2.7.4 Manhattan Distance

The sum of the absolute differences Eq. (2.6) between the Cartesian coordinates of two vectors [303], x and y in an n-dimensional real vector space determines the Manhattan Distance.

$$M(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{2.6}$$

In the case where x and y are identical, $M(x,y) = 0$ would equate to 0, if not, M (x, y) > 0. Similarly, as defined in Eq. 2.1, for the Euclidean distance, the Manhattan distance could be normalised to be within values 0 and 1 for vectors whose coordinates are non-negative real numbers less than or equal to 1. Furthermore, the derivation of Eq. (2.7) results in a similarity matrix as below.

$$m(x,y) = \frac{1}{n} \sum_{i=1}^{n} |x_i - y_i| \tag{2.7}$$

Likewise, a similarity metric can be derived from Eq. (2.8) as follows.

$$Sim_m(x,y) = 1 - m(x,y) \tag{2.8}$$

2.8 Summary

This chapter has presented an extensive discussion of the literature review of this thesis. The focal point of this chapter discussed the relevant benefits, challenges, and theories related to the concept of adaptive e-Learning recommender systems. Moreover, five main aspects and background information related to the concepts of adaptive e-learning, learning styles, student profile adaptation, recommendation algorithms and clustering in the e-learning system have also been discussed upon.

Through this literature review, the research gap or limitations related to the relevant field of study which are necessary to be addressed in order to develop personalised e-learning systems are revealed. These limitations are known to be focused around the following points: first, student learning styles initialisation to overcome the cold start problem; second, mechanisms used to capture and interpret student learning styles from learning behaviour patterns; third, techniques used to recommend learning objects according to student adaptive profile.

Nonetheless, a great deal of work has been conducted on adaptive e-learning recommendation systems, yet there is no comprehensive solution which successfully incorporates the above-mentioned problems; some solutions concentrate on solving cold-start problems to improve the accuracy of LOs recommendations, others focused on determining student learning styles.

Hence, in this thesis, we aim to overcome the aforementioned limitations by proposing a novel adaptive e-learning recommender system aid students in finding appropriate learning materials. This will be achieved by building a dynamic student profile (learning styles) and developing a new algorithm for recommending learning objects to suit individual student learning styles. Additional explanation and details about the proposed architecture will be presented in Chapter 3.

Chapter 3

Ubiquitous LEARNing system (ULEARN) Architecture

The effectiveness of Adaptive E-Learning Recommender Systems (AERSs) depends on their ability to suggest appropriate learning content in line with learners' Learning Styles and preferences. As mentioned in this thesis' literature review (Chapter 2), existing AERSs still face several challenges, some of which are: 1) how to initialise the Learning Style of a new student; 2) how to recommend learning materials according to Learning Styles of students, and 3) how to identify student's Learning Styles from their learning behaviour patterns.

Primarily, this chapter introduces a novel adaptive e-learning recommendation (ULEARN) framework to recommend personalised learning objects according to students' Learning Styles. The major innovative features of ULEARN include: (i) A novel algorithm for initialising the student Learning Style using a dynamic Learning Style questionnaire; (ii) An innovative algorithm for recommending learning objects according to learners' Learning Styles; (iii) An algorithm for dynamically adapting student Learning Styles based on learning behaviour patterns. The proposed ULEARN framework will help in answering research question 1:

RQ1. What is an appropriate architecture for an adaptive learning object recommender system?

The remainder of this chapter is structured in the following way. The first section reviews the key concepts of the proposed adaptive e-learning recommendation framework while the proposed ULEARN architecture and its layers have been presented to the reader in Section 3.2. Next, Section 3.3 presents the computational model of the framework in addition to Section 3.4 which provides two real examples to the reader. Lastly, a brief overview of this chapter is described in Section 3.5.

3.1 Proposed adaptive e-learning recommendation

Through the work in this thesis, we propose a novel adaptive e-learning recommendation architecture for recommending personalised learning objects on the basis of student learning styles. The ULEARN architecture aims to overcome issues related to cold-start and rating sparsity by incorporating learning object profiles in addition to students' learning styles.

As already mentioned in Chapter 2 of this thesis, the FSLSM Learning Style model has served as the basis for our proposed framework. The four dimensions (Information Processing *(Active/Reflective)*, Information Perception *(Sensing/Intuitive)*, Information Input *(Visual/Verbal)*, and Information Understanding *(Sequential/ Global)*) of the FSLSM model are made use of for the representation of student Learning Styles as well as learning object profiles.

Therefore, we will briefly introduce the concepts of student profile, object profile and course structure, which served as the basis of our architecture, before describing the proposed architecture itself. The following sub-sections will describe these profiles in detail.

3.1.1 Student Profile

Student Profile (SP) is the key resource for facilitating our proposed system to represent essential information about each student's preferences. The purpose of the SP in the proposed system is to store the student's Learning Style. Every student has their own profile which allows the system to provide a personalised learning experience in accordance with their learning styles.

A student's Learning Style is comprised of a vector of real values that ranges between 0 to 1 (or from 0% to 100%), as below in Eq. (3.1), where Learning Style attributes are used as place holders.

$$\textbf{LS = (Active, Reflective, Visual, Verbal, Sequential, Global, Sensing, Intuitive)} \tag{3.1}$$

Example 1. Demonstrated below in Table 3.1 are few examples of student Learning Style vectors from ULEARN database (See Sect.7.3) which are determined by making use of the learner's responses to the ILS questionnaire [304] or by their interaction with the system [278].

Table 3.1: Examples of student learning style vectors

	Student profile							
	Active	**Reflective**	**Visual**	**Verbal**	**Sequential**	**Global**	**Sensing**	**Intuitive**
Yousef	0.5	0.5	0.6	0.4	0.8	0.2	0.7	0.3
Clara	0.4	0.6	0.15	0.85	0.7	0.3	0.8	0.2
Ed	0.5	0.5	0.6	0.4	0.8	0.2	0.7	0.3
George	0.5	0.5	0.65	0.35	0	1	0.55	0.45

3.1.2 Learning object Profile

Learning objects for a specific lesson are generated by the organisation of learning content materials. In an e-learning environment, a learning object (LO) is an indispensable tool used for the preparation of tutorials for learners in electronic and reusable formats such as lecture notes, presentations, questions, activities, examples, exercises, etc. In this research, LOs are presented in numerous formats, namely, text files (pdfs), presentations (PowerPoint slides), pictures and videos, to suit the Learning Styles of individual learners.

For instance, a visual learner would favour watching a video rather than reading a pdf document, compared to a verbal learner who would prefer doing exactly the opposite. Thus, an FSLSM vector can be used to represent a learning Object Profile (OP) to indicate the class of learners that the respective learning object is appropriate for, as shown in Eq. (3.2).

$$\textbf{OP = (Active, Reflective, Visual, Verbal, Sequential, Global, Sensing, Intuitive)} \tag{3.2}$$

It is important to note that unlike student Learning Styles which are computed by means of the ILS questionnaire or learning behaviour patterns, it is generally presumed that the learning object profile is created by the teacher or an educational professional.

Example 2. For illustration purposes, few examples of OPs stored in ULEARN database (See Sect. 7.3) created by the teacher presented in Table 3.2.

3.1.3 Course structure

Within this section, we will introduce the course structure, which is an essential component in personalising course recommendation. In this research, we focused on the course structure according to Egyptian

Table 3.2: Examples of learning object profile

	Object profile							
	Active	**Reflective**	**Visual**	**Verbal**	**Sequential**	**Global**	**Sensing**	**Intuitive**
LO_1	0.7	0.3	0.5	0.5	1	0	0.3	0.7
LO_2	0.2	0.8	0.8	0.2	0.7	0.3	0.4	0.6
LO_3	0.5	0.5	0.2	0.8	0.1	0.9	0.6	0.4
LO_4	0.5	0.5	0.5	0.5	0.5	0.5	0.3	0.7
LO_5	0.9	0.1	0.7	0.3	0.3	0.7	0.1	0.9

higher education system. The structure of the course material comprises three levels, namely, courses, lessons and learning objects. While each course is grouped into several lessons; each lesson is composed of a set of learning objects in several formats such as text, video, audio, etc, as shown in Figure 3.1.

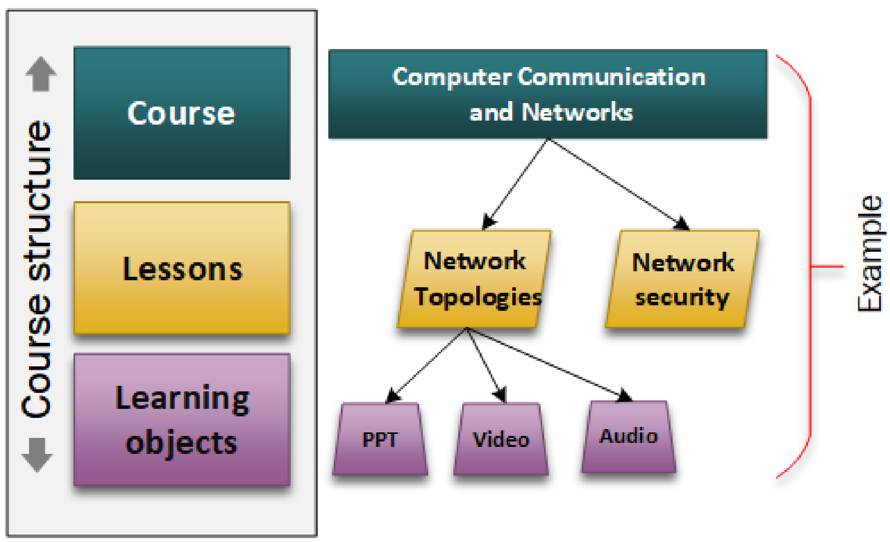


Figure 3.1: Structure of the course materials in Egyptian higher education system

The next section describes the proposed adaptive e-learning recommendation technique as well as the components in more detail.

3.2 Architecture of ULEARN

This section introduces an overview of the proposed conceptual framework for adaptive e-learning recommendation systems. The proposed framework comprises three layers: *'Presentation layer'*, *'Adaptive-recommendation layer'* and *'Data layer'*, as illustrated in Figure 3.2. Each layer contains several key components that interact with one another and with components in other layers. A crucial layer, the

Adaptive-recommendation Layer comprises the core of ULEARN adaptive e-Learning framework as explained below and presented in the architectural framework design in Figure 3.2.

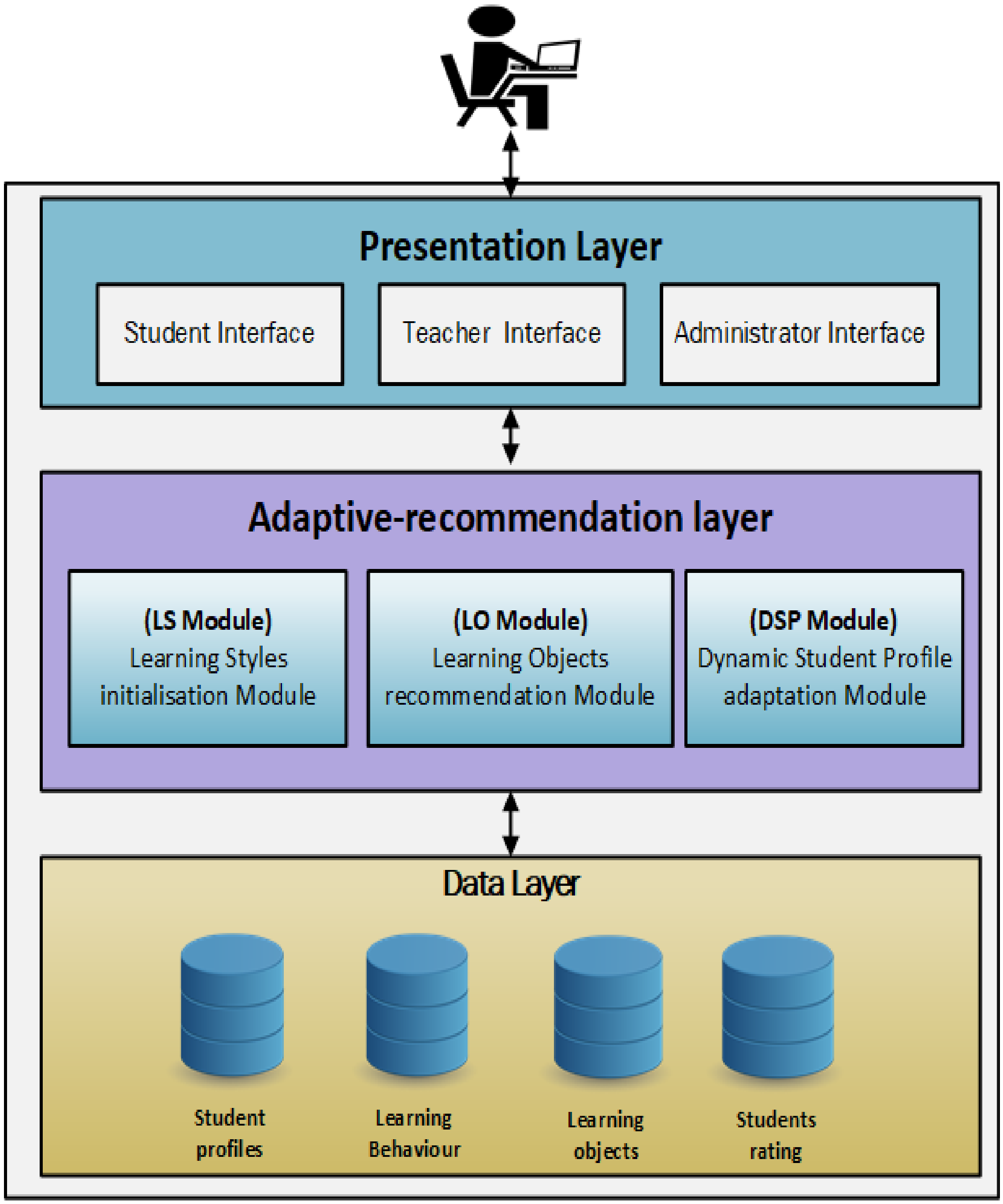


Figure 3.2: Conceptual model of Ubiquitous LEARNing system (ULEARN)

The following sections illustrate the three main layers of the proposed recommendation system and show how they interact together.

3.2.1 Presentation Layer

A presentation layer permits interaction between the user and the system, while also being responsible for communicating with other layers, by which it: 1) passes relevant information onto the adaptive-recommendation layer and data layer, and 2) presents the outcomes of the interaction with the internal framework layers and module. This presentation layer functions as the interface through which the user accesses the system. A Graphical User Interface (GUI) contains the Student Interface, Teacher Interface as well as the Administrator Interface, as illustrated in Figure 3.2.

- **Student Interface:** Concerned with the student's account system (such as registration and login). The process begins when the student uses the ULEARN portal to complete registration followed by which personal details including the student's full name and email address are gathered and saved into the student profile database. This interface is also responsible for displaying learning objects recommended by the system in addition to the collection of student ratings.

- **Teacher Interface:** Allows teacher to manage course lessons, create and modify course learning objects in different formats to match student Learning Styles, as well as monitor students' performances based on all types of assignments, and grading.

- **Administrator Interface:** Allows the administrator to assign teachers to specific courses in addition to managing the system database.

3.2.2 Data Layer

The ability to store and access data is provided by the data layer (for example, API with respect to the proposed e-Learning Framework). This interface at the bottom of the three-tier framework where a physical SQL database is hosted and stores all data related to the student's profile, learning object profile, and learning behaviour patterns.

- **Students profiles database:** Stores both static and dynamic data regarding the learner. Details such as full name, email address, password of the learner constitute static data while dynamic data includes student Learning Styles. This data is updated, according to the student's learning behaviour patterns.

- **Learning behaviour database:** Stores data about student's interaction with the system including time spent for learning in addition to the specific learning objects that were visited (detailed discussion in Chapter 6). This data will be used later for updating student Learning Styles.

- **Learning object database:** Stores details related to learning object formats, total time and profiles (Learning Styles). The learning object's profile is retrieved by the Learning Objects recommendation Module to calculate the similarity between the student's Learning Style and Learning Objects.

- **Student rating database:** Stores history of the student's rating given to LOs they have previously studied. Each rating is associated with the student_ID, the LO_ID, the rating value (ranging from 1 to 5), and the timestamp.

3.2.3 Adaptive-recommendation layer

As mentioned earlier, the adaptive-recommendation layer which constitutes the main capabilities of the ULEARN e-Learning framework (as shown in Figure 3.2) consists of the following three modules:

- Learning Styles initialisation Module *(LS Module)*;

- Learning Objects recommendation Module *(LO Module)*;

- Dynamic Student Profile adaption Module *(DSP Module)*.

Each module is responsible for processing a specific step in order to work together as an integrated adaptive e-learning recommendation system. This integration helps to build a solid foundation for supporting an adaptive e-learning environment.

3.2.3.1 Learning Styles initialisation Module

As discussed in Chapter 2, the ability of an e-learning recommender system to recommend suitable learning content in line with a learner's Learning Style and preferences demonstrates its effectiveness. For this purpose, building of an efficient learner profile in order to acquire personalisation and adapt to various styles constitutes a constructive approach. It is generally essential that the learner's Learning Style and preferences on a domain are made known prior to adapting the learning process and course content. Initialisation of a student's Learning Style is considered one of the main requirements for the adaptation process to take place. The adaptation process begins based on this step to overcome issues related to a cold-start, which constitutes the issue of acquiring relevant data for building an initial profile.

The *LS Module* uses the dynamic Learning Style questionnaire for focusing on the initialisation of an adaptive learner profile. Following the registration process, the student's Learning Styles are tested through their responses to the dynamic variant of the questionnaire.

Furthermore, an innovative feature of the *LS Module* comprises its potential to ascertain a learner's Learning Style in every dimension using the student's responses to just a few questions of the questionnaire; thus saving significant student time and effort from having to answer all 44 questions within the FSLSM Learning Style questionnaire in addition to the accuracy of Learning Styles identification. The initial student Learning Styles are stored within the student profile, to be used for recommending appropriate learning objects to learners for increasing their learning and overcoming the cold-start problem. Further explanation and details related to the proposed algorithm will be presented in Chapter 4.

3.2.3.2 Learning Objects recommendation Module

The recommendation module is responsible for generating and delivering the top_n learning objects list to students based on their learning style stored in their profile. We proposed three different algorithms for recommending personalised learning objects list, namely: Enhanced Collaborative Filtering *(ECF)*, Enhanced Content-Based Filtering *(ECBF)* and Enhanced Hybrid Filtering *(EHF)*. Unlike traditional approaches, each of these algorithms handles the cold-start and the rating sparsity problems effectively using information from the students learning styles and the learning objects profiles in addition to rating values in its recommendation process.

The results from the experiment inferred that the EHF algorithm has the best LOs recommendation with respect to the accuracy of rating predictions. The proposed *EHF* algorithm is a combination of *ECF* and *ECBF*. An innovative feature regarding the three recommendation algorithms comprises establishing a relationship between the student profile, learning objects profile as well as student rating, to tackle the issuess of cold-start and rating sparsity and improve the recommendation accuracy. An explanation of the detailed working of the recommender algorithm is provided in Chapter 5.

3.2.3.3 Dynamic Student Profile adaptation Module

A critical problem of building an adaptive e-learning environment comprises the tedious task of discovering a proper way to dynamically model student preferences and track their changes, since students' preferences and abilities are constantly changing. The *DSP Module* aims to build and update student profiles and Learning Styles frequently. The student profiles are automatically updated based on their learning behaviour patterns while studying the recommended learning objects. This behavioural data which is stored in the database is obtained from sessions of the students' interaction with the system (see Section 3.2.2).

The *DSP Module* makes use of a novel methodology to allow the extraction of features in order to characterise student behaviour for identifying student Learning Styles with respect to the FSLSM, which are then utilised to dynamically update student profiles as follows:

- Extraction of student learning behaviour patterns which reflect Learning Styles from system log-file.
- Calculation of student Learning Styles.
- Updating of student Learning Styles dynamically after a number of sessions.

Chapter 6 presents how students' Learning Styles are updated dynamically in details.

3.3 Computational Model of ULEARN

As mentioned previously (see Section 3.2.3), the proposed adaptive e-learning recommendation process constitutes three major modules which are; *LS Module*, *LO Module*, *DSP Module*, as shown in Figure 3.3. The process begins with the learner's interaction with the ULEARN. When students log into the system for the first time, they are requested to fill-out the dynamic LS questionnaire so as to initialise their Learning Styles profile. The *LO module* then selects the most appropriate LOs according to the student's Learning Styles. First, the LO module retrieves available learning object profiles from the database, then applies the EHF algorithm to compute the similarity between LOs Profiles and the active student's Learning Style. After calculating this similarity, the system recommends top-n LOs list in a specific topic in accordance with the active student's Learning Styles.

Once the student has completed studying the learning objects, it is necessary to update the Learning Styles on the basis of their learning behaviour patterns as presented in the last section of the flowchart. In this step, the *DSP Module* retrieves the student's learning behaviour patterns from system log-file to calculate the new Learning Styles of the student taking into consideration their old profile. Subsequently, the new Learning Styles will be saved in the student's profile database. Finally, when the student logs into the system, the new list of learning objects will be recommended according to their updated Learning Styles.

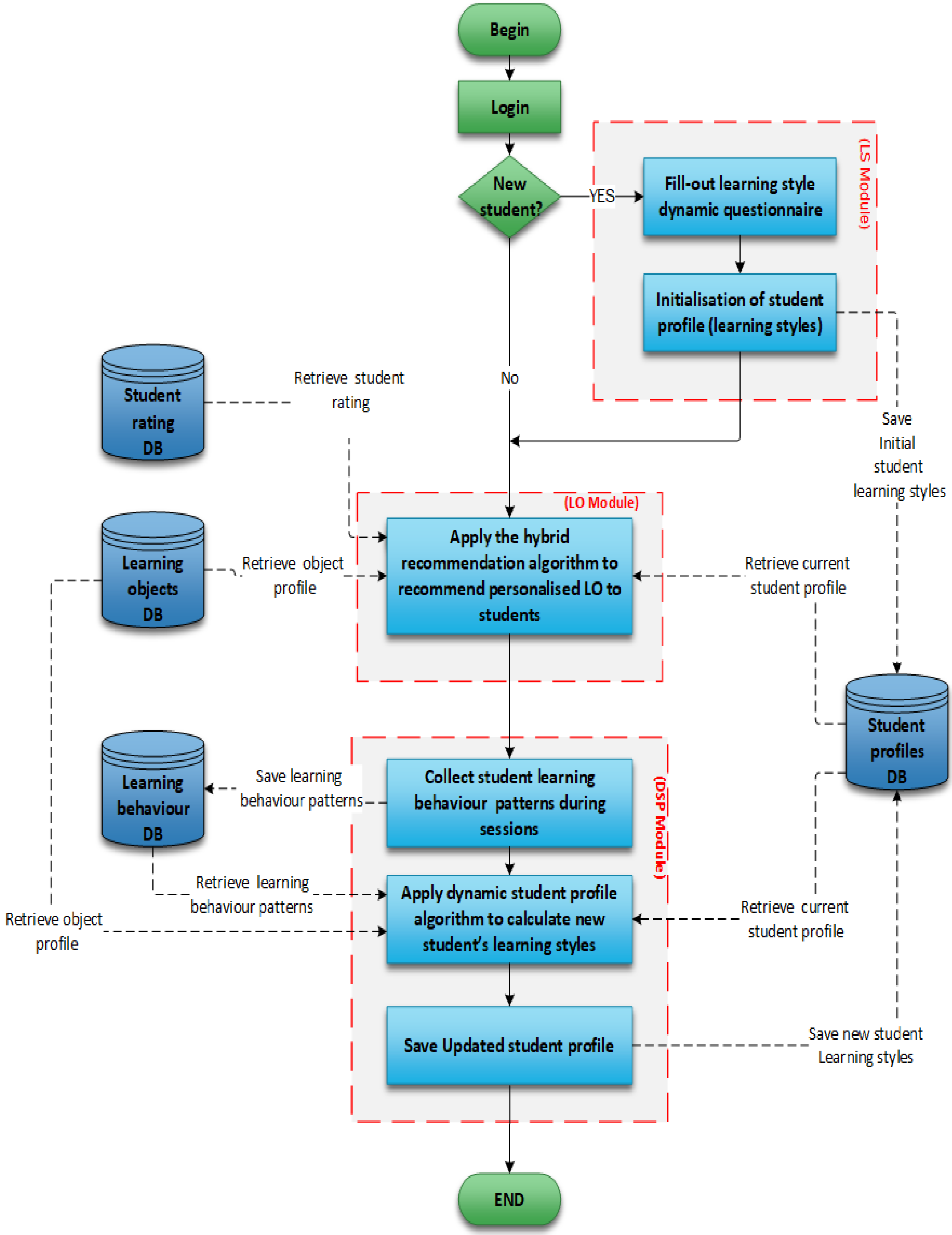


Figure 3.3: Computational Model of ULEARN

In the next section, two distinct scenarios have been presented to the reader to illustrate the above workflow and how the three modules are integrated into one system.

3.4 Examples

To clearly interpret the proposed framework, here, we present two scenarios based on a real academic case study, thus showing the entire process of adaptation which contains three steps, as shown in Figure 3.3. The students participated voluntarily in the study from Arab Academy for Science and Technology (AAST).

3.4.1 First scenario (New students)

Step 1*– Learning style initialisation.* The process begins with two new students Emma and Oliver who carry out the learning activity by interacting with the ULEARN. The system then asks them to register using their personal details (full name, email address, password, and major), to fill-out the dynamic LS questionnaire so as to initialise their profiles (Learning Styles), as presented in Table 3.3 (See Section 3.2.3.1). Following this, the system saves Emma's and Oliver's Learning Styles in the student profile database.

Table 3.3: Emma's and Oliver's initial profile

	Initial Student's profile(learning styles)							
Students	**Active**	**Reflective**	**Visual**	**Verbal**	**Sequential**	**Global**	**Sensing**	**Intuitive**
Emma	**0.7**	**0.3**	**0.2**	**0.8**	**0.5**	**0.5**	**0.6**	**0.4**
Oliver	**0.4**	**0.6**	**0.1**	**0.9**	**0.7**	**0.3**	**0.8**	**0.2**

Step 2*–Learning objects recommendation.* As seen in Step 1, Emma and Oliver are new in the system thus inferring that they do not have any rating on LOs. Therefore, the system delivers recommendations by calculating similarities between the students' Learning Styles and object profile. The proposed algorithm retrieves students' current Learning Styles, as well as available objects profiles from the database in order to find the appropriate learning objects. Table 3.4 demonstrates examples of the recommended learning objects.

Once they finish the learning objects, the system requests them to rate the LOs from 1 to 5 in order to use this rating while calculating predicted ratings in the upcoming sessions.

Table 3.4: Recommended learning object profile according to Emma's and Oliver's initial profiles

Student profiles	Recommend LOs Types	Learning Object Profiles							
		Active	Reflective	Visual	Verbal	Sequential	Global	Sensing	Intuitive
Emma's SP	video	0.65	0.35	0.2	0.8	0.4	0.6	0.55	0.45
	Summary	0.2	0.8	0.3	0.7	0.2	0.8	0.6	0.4
Oliver's SP	PDF	0.4	0.6	0.1	0.9	0.7	0.3	0.8	0.2
	PPT	0.5	0.5	0.6	0.4	0.5	0.5	0.5	0.5

Step 3*–Dynamic student profile adaption.* In this step, the learning behaviour patterns of Emma and Oliver are collected in the course of the number of sessions, including the time spent on each learning object (LO), the type of LOs (e.g. image, PowerPoint, or pdf) accessed, etc. Once the learning behaviours are collected, the dynamic profile algorithm calculates new Learning Styles of Emma and Oliver taking into account their current Learning Styles. Subsequently, Emma's and Oliver's profiles are updated with the latest Learning Styles (see Table 3.5).

Table 3.5: Emma's and Oliver's new learning styles profile

	New profile (learning styles)							
Student	Active	Reflective	Visual	Verbal	Sequential	Global	Sensing	Intuitive
Emma	0.52	0.48	0.2	0.8	0.5	0.5	0.6	0.4
Oliver	0.5	0.5	0.1	0.9	0.7	0.3	0.8	0.2

Finally, the proposed framework will make use of the new Learning Styles to recommend new learning objects when they log in again into the system, as explained in the following scenario.

3.4.2 Second scenario (Existing students)

Step 1*– Retrieve student's profiles.* When Emma and Oliver log into the system again, the system attempts to retrieve their new Learning Styles (according to their learning behaviour patterns) as well as their rating on recommended learning objects, as shown in Tables 3.5 and 3.6 (See Section 7.1.4). Once this information is collected, the system then sends a request to the *LO Module* to suggest the new LOs list according to the new Learning Styles.

Step 2*–Learning objects recommendation.* Based on the information retrieved from Step 1, the recommender learning object module starts to apply the hybrid algorithm taking into consideration

Table 3.6: Emma's and Oliver's learning objects Ratings

Student	Recommend LOs Types	Rating
Emma's rating	video	⋆⋆⋆
	Summary	⋆⋆⋆⋆
Oliver's rating	PDF	⋆⋆⋆⋆⋆
	PPT	⋆⋆

Learning Styles of students, objects profiles along with rating information to recommend the new LOs list. Table 3.7 demonstrates the new LOs recommended list based on the updated Learning Styles.

Table 3.7: The new learning object recommended list according to Emma's and Oliver's new learning styles

Student profiles	Recommend LOs Types	Learning Object Profiles							
		Active	Reflective	Visual	Verbal	Sequential	Global	Sensing	Intuitive
Emma's SP	Video	0.5	0.5	0.75	0.25	0.75	0.25	0.6	0.4
	Book Chapter	0.2	0.8	0.2	0.8	1	0	0.5	0.5
Oliver's SP	Audio	0.3	0.7	0.1	0.9	0.8	0.2	0.7	0.3
	PDF	0.25	0.75	0.2	0.8	0.75	0.25	0.5	0.5

Step 3*–Dynamic student profile adaption.* Once Emma and Oliver have finished these learning objects, the proposed framework will gather their learning behaviour patterns again (as illustrated in Step 3 of the first scenario) to update their profiles again, as shown in Table 3.8.

Table 3.8: Emma's and Oliver's updated learning style learning styles profiles (Second scenario)

	New profile (learning styles)							
Student	Active	Reflective	Visual	Verbal	Sequential	Global	Sensing	Intuitive
Emma	0.5	0.5	0.3	0.7	0.4	0.6	0.65	0.35
Oliver	0.4	0.6	0.2	0.8	0.65	0.35	0.7	0.3

Each of the above steps in these scenarios will be illustrated step-by-step, with calculations in the coming chapters. The following section summarises and concludes this chapter.

3.5 Summary

This chapter has presented an overview of the proposed ULEARN adaptive e-learning recommendation architecture. In particular, a three-tier architecture that separates the presentation layer from the adaptive-recommendation layer and the data layer has been put forth. The proposed architecture modules aim to address questions associated with the initialisation of Learning Styles, automatic identification of Learning Styles from student learning behaviour patterns as well as learning objects recommendation in line with student's Learning Styles. A computational model of the framework workflow was used to show the interactions between the three modules. Moreover, the proposed architecture is a novel, personalised, adaptive dynamic hybrid recommender framework which provides a solution to issues related to information overloading, cold-start and rating sparsity pertaining to the difficulty of providing a high recommendation quality. The next chapter discusses the initialisation of student Learning Styles based on a dynamic ILS questionnaire algorithm in detail in addition to an empirical study which determines the question order for every dimension of the FSLSM Learning Style questionnaire.

Chapter 4

Dynamic Learning Styles Questionnaire for Student Profile Initialisation

Acquiring information about student learning styles is an important task since it comprises the initial step in building an adaptive e-learning recommender system which in turn is required to develop a student profile containing the personal preferences of the student. The challenge in this context is how to obtain learning styles of new students when the systems have insufficient information with which to generate high-quality personalised recommendations. This is commonly referred to as the "cold- start" problem.

This chapter focuses on initialising students learning styles by making use of the FSLSM questionnaire based on learning styles and proposes a novel algorithm to initialise students' learning styles using a dynamic ILS questionnaire by having them answer a reduced number of questions compared to that of the FSLSM model in order to identify their learning styles. Once filled in, the answers to this recently developed questionnaire, which is reorganised into four main categories - one per learning style dimension, is then used by a novel algorithm, which is constructed upon the latest ranking of the questions to determine a student's initial learning styles. A study has been conducted so as to ascertain the order in which questions are asked in each dimension.

Thereby in this chapter, we attempt to work on and find an appropriate solution for the second research question:

RQ2. What is the best way to initialise student learning styles?

The remainder of this chapter is organised as follows. First, Section 4.1 introduces the reader to an algorithm which is used to create a learner profile followed by which Section 4.2 presents an overview on the Index of Learning Styles as made use of by the algorithm. Section 4.3 comprises an empirical study to rank the questions within the questionnaire while Section 4.4 provides the algorithm to compute the Learning Style of the user. Finally, a real-time example of the model has been presented in Section 4.5 in addition to a chapter summary in Section 4.6.

4.1 Algorithm for Initialising Student Learning Styles

This section presents the proposed architecture for initialising student learning styles through a dynamic learning style questionnaire algorithm and is illustrated in Figure 4.1.

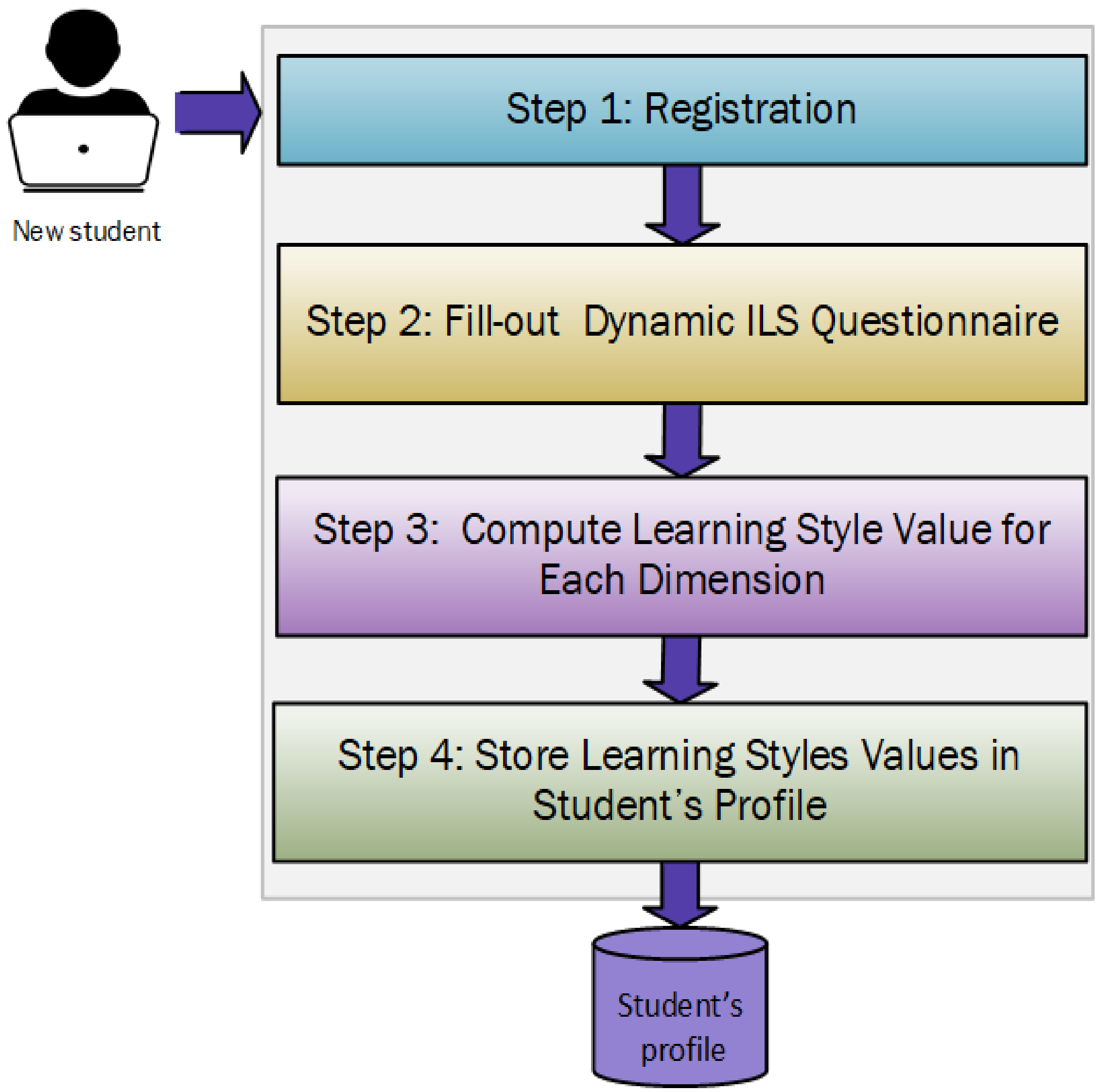


Figure 4.1: Initialising Student Learning Styles Architecture

As described in Section 3.2.1, the student profile including the personal details is gathered from a student during the registration process, followed by which the student is required to fill out a dynamic ILS questionnaire. This proposed architecture of student learning styles initialisation comprises the following:

- ***Step 1: (Registration)*** A student uses the ULEARN learning portal to register themselves in advance. During the registration process, personal details of the user including the user's name, email address and password are gathered.

- ***Step 2: (Fill-out learning style questionnaire)*** Following registration, the student fills-out the LS questionnaire. As the student goes through the questionnaire, the system dynamically computes their learning style for every dimension by counting the number of answers "a" and the number of answers "b". Once the number of "a" (or "b") reaches 7 (i.e. 60% of the 11 questions) within one dimension, the system skips the remaining questions for that specific dimension and moves to questions within the next dimension.

- ***Step 3: (Compute learning style value for each dimension)*** Computing every dimension's learning style as the percentages of "a" and "b". For instance, in the dimension *information input*, one may possess preferences related to 60% visual and 40% verbal.

- ***Step 4: (Store learning style values in the student's profile)*** Initial Learning Styles computed through the dynamic ILS questionnaire is respectively stored within the student profile database.

The algorithm of the proposed dynamic LS questionnaire comprises two major components:

- First, the offline working of an empirical study is made use of to ascertain the questions sequence for each of the four dimensions within the Felder-Silverman Learning Style questionnaire.

- Second, an algorithm is constructed in line with the question ranking to dynamically compute students' initial Learning Styles as they progress through the questionnaire.

Before we describe the proposed dynamic Learning Styles questionnaire, it is crucial to briefly introduce the Index of Learning Styles questionnaire. Following this, the empirical study for purpose of ranking the questions within respective questionnaire is described in detail.

4.2 Index of Learning Styles

Comprising 44 questions, eleven in each dimension, the Index of Learning Styles (ILS), developed by Felder and Silverman is used for the identification of Learning Styles in line with the Felder-Silverman

Learning Style model [2].

As previously mentioned in Chapter 2, every individual learner possesses a personal preference with respect to each dimension, expressed within values between +11 to -11 per respective dimension. This range comes from the 11 questions posed for each dimension. While answering a question for example, 1 is added to the value for an answer related to the "active" preference of the *active/reflective* dimension while 1 is subtracted from the value for a reflective preference.

Hence, the answer to every question is either +1 (answer ”a”) or -1 (answer ”b”). However, in the case where both ”a” and ”b” could apply, it is necessary for users to select the answer which applies more frequently to them. For instance, a sample question could be: I understand something better after I, a) try it out; or b) think it through.

Measuring student’s Learning Styles using questionnaires results in many drawbacks. While it is noted that the FSLSM-ILS questionnaire could be long for some students, it could also lead to uncalled-for behaviours including skipping questions, false answering, or even giving up the questionnaire (in addition to the entire system) altogether [128–131]. Consequently, e-learning systems cannot accurately capture student Learning Styles, thus leading to poor recommendations. It also seems very time-consuming for a student to fill-out the questionnaire.

The following section presents an overview of the empirical study conducted for the purpose of ranking questions within the questionnaire.

4.3 Ranking the Questionnaire Questions

This section describes the processes of ranking questions within the ILS questionnaire. The proposed dynamic algorithm is significantly different from the algorithm and saves time and effort by not having to go over all of the 44 questions within the questionnaire for determining the user’s Learning Style and preferences. For this, the ILS questionnaire has been restructured into four groups of 11 questions for every learning style dimension of the FSLSM. The straightforward questions come first in each dimension, and then the ambiguous (or unclear) ones last, according to how convenient it is to select a singular answer between options “a” and “b”.

Since our goal comprised the enhancement of an accurate identification of learning styles by reducing the number of questions asked, consequently, an experimental study was conducted in order to infer the ranking of questions within every dimension.

4.3.1 Participants

Involvement in the study included voluntary participants from the Arab Academy for Science and Technology (AAST), Egypt with a target sample size of 80 students randomly selected from three different majors (Business Information Systems, Accounting and International Market).

As part of the study, the participants were given a task to indicate the ease of choosing between answer options "a" and "b" for every question in the ILS questionnaire. For this purpose, a 5-level Likert scale was made use of where 1 and 2 respectively indicated Very easy and Easy, while a score of 3 indicated Intermediate and 4 and 5 indicated Difficult and Very Difficult respectively.

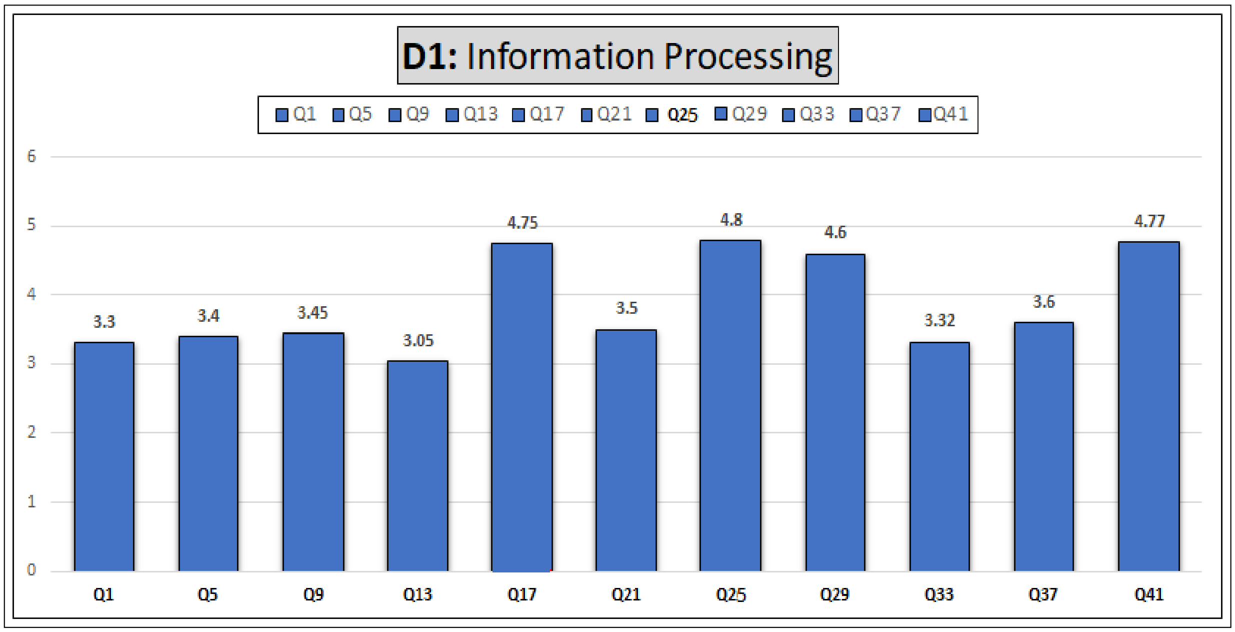


Figure 4.2: Difficulty level of questions in Information Processing

Out of the 80 responses which were collected, 75 were made use of for analysing purposes while the remaining five weren't included since they were not complete. The score for every question was totalled in addition to being normalised, the results of which are demonstrated in Figs. 4.2 to 4.5. **D1** to **D4** corresponds to the *active/reflective*, *sensing/intuitive*, *visual/verbal* and *sequential/global*, respectively.

The results showed that some questions in the ILS questionnaire were more difficult than others within the same dimension. Hence, it was decided to ask the straightforward questions first to correctly detect student's Learning Styles. Ambiguity in questions leads to lower data quality and incorrect student Learning Styles identification.

As presented in Tables 4.1 to 4.4, questions have been ranked in an ascending order of difficulty levels within each dimension with respect to the results of this study.

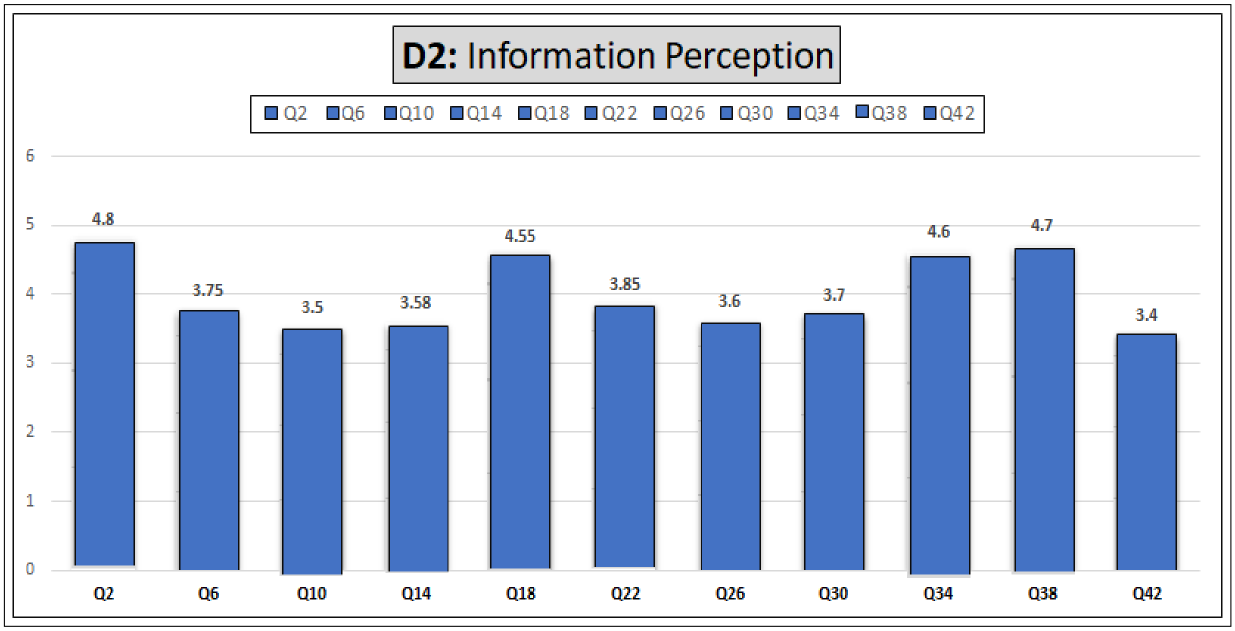


Figure 4.3: Difficulty level of questions in Information Perception

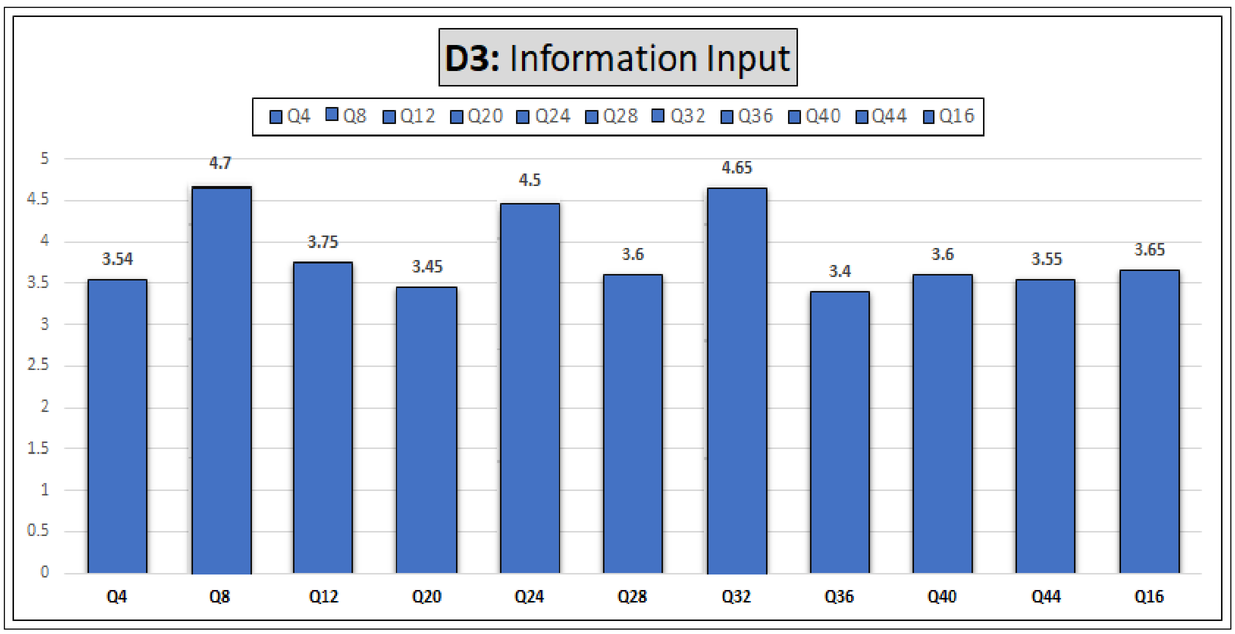


Figure 4.4: Difficulty level of questions in Information Input

Based on the new ranking of ILS questionnaire's questions, the algorithm starts to calculate student's initial Learning Styles, as described in the following section.

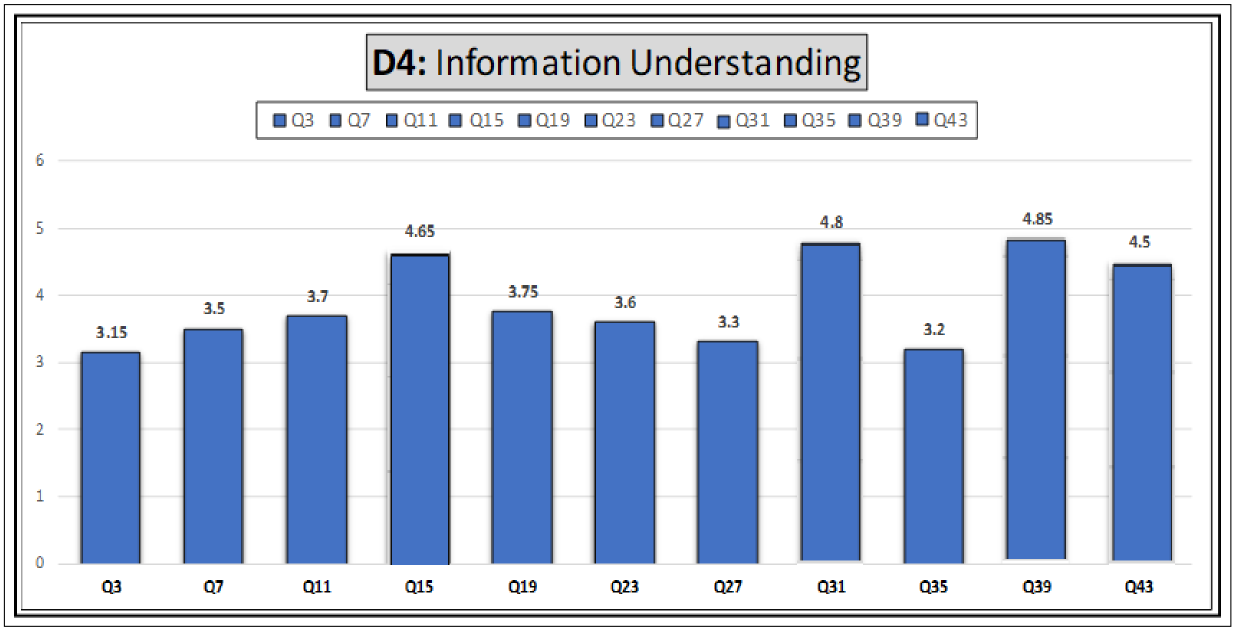


Figure 4.5: Difficulty level of questions in Information Understanding

Table 4.1: Order of Questions in Information Processing Dimension

Questions	FSLSM Question's Sequence	ULEARN Question's Sequence	Difficulty level
D1: Information processing (Active/ Reflective)			
In classes I have taken: (a) I have usually gotten to know many of the students. (b) I have rarely gotten to know many of the students	13	1	3.05
I understand something better after I: (a) try it out. (b) think it through	1	2	3.3
When I have to work on a group project, I first want to: a) have "group brainstorming" where everyone contributes ideas. (b) brainstorm individually and then come together as a group to compare ideas	33	3	3.32
When I am learning something new, it helps me to: (a) talk about it. (b) think about it	5	4	3.4
In a study group working on difficult material,I am more likely to: (a) jump in and contribute ideas. (b) sit back and listen.	9	5	3.45
I prefer to study: (a) in a study group. (b) alone.	21	6	3.5
I more easily remember: (a) something I have done. (b) something I have thought a lot about.	29	7	3.6
I am more likely to be considered: (a) outgoing. (b) reserved.	37	8	3.6
When I start a homework problem, I am more likely to: (a) start working on the solution immediately. (b) try to fully understand the problem first	17	9	4.75
The idea of doing homework in groups, with one grade for the entire group: (a) appeals to me. (b) does not appeal to me	41	10	4.77
I would rather first (a) try things out. (b)think about how I'm going to do it.	25	11	4.8

Table 4.2: Order of Questions in Information Perception Dimension

Questions	FSLSM Question's Sequence	ULEARN Question's Sequence	Difficulty level
D2: Information perception (Sensory/ Intuitive)			
When I am doing long calculations: a) I tend to repeat all my steps and check my work carefully. b) I find checking my work tiresome and have to force myself to do it.	42	12	3.4
I find it easier: a) to learn facts. b) to learn concepts.	10	13	3.5
In reading nonfiction, I prefer: (a) something that teaches me new facts or tells me how to do something. (b) something that gives me new ideas to think about	14	14	3.58
When I am reading for enjoyment,I like writers to: (a) clearly say what they mean. (b) say things in creative, interesting ways	26	15	3.6
When I have to perform a task, I prefer to: (a) master one way of doing it. (b) come up with new ways of doing it	30	16	3.7
If I were a teacher, I would rather teach a course: (a) that deals with facts and real life situations. (b) that deals with ideas and theories	6	17	3.75
I am more likely to be considered: (a) careful about the details of my work. (b) creative about how to do my work.	22	18	3.85
I prefer the idea of (a) certainty. (b) theory	18	19	4.55
I consider it higher praise to call someone : (a)sensible. (b) imaginative.	34	20	4.6
I prefer courses that emphasise: (a) concrete material (facts, data). (b) abstract material (concepts, theories).	38	21	4.7
I would rather be considered: (a) realistic. (b) innovative.	2	22	4.8

4.4 Proposed algorithm for calculating Initial Learning Style

The pseudocode as illustrated in Fig. 4.6 and the flowchart in Fig. 4.7 briefly summarise the algorithm `Initial_LS` which computes a learner's initial Learning Style using the new ordering of questions within the ILS questionnaire (Tables 4.1 to 4.4). This algorithm sums up answers to options "a" or "b" for all eleven questions within each dimension followed by computing the difference between these values to arrive at a decision. Once the number of answers to options "a" or "b" becomes 7, the remainder of the questions within that specific dimension are skipped.

As mentioned earlier, a learner with a score of 1 or 3 (-1 or -3) demonstrates a mild preference for X (resp. for Y); however, is quite well-balanced to be able to positively learn in an environment that favours X or Y. A learner with a score of 5 or 7 (-5 or -7) has a moderate preference for X (resp. for Y) and would learn more easily in an environment which favours X (resp. Y). Lastly, a score of 9 or 11 (-9 or -11) demonstrates a strong preference for X (resp. for Y); and these learners could face great challenges

Table 4.3: Order of Questions in Information Input Dimension

Questions	FSLSM Question's Sequence	ULEARN Question's Sequence	Difficulty level
D3: Information input (Visual/Verbal)			
When I am learning a new subject, I prefer to: a) stay focused on that subject, learning as much about it as I can. b) try to make connections between that subject and related subjects.	36	23	3.4
I tend to; a) understand details of a subject but may be fuzzy about its overall structure. b) understand the overall structure but may be fuzzy about details	4	24	3.45
It is more important to me that an instructor: a) lay out the material in clear sequential steps. b) give me an overall picture and relate the material to other subjects.	20	25	3.45
When solving problems in a group, I would be more likely to: a) think of the steps in the solution process. b) think of possible consequences or applications of the solution in a wide range of area.	44	26	3.55
When considering a body of information, I am more likely to: a) focus on details and miss the big picture. b) try to understand the big picture before getting into the details.	28	27	3.6
Some teachers start their lectures with an outline of what they will cover. Such outlines are: (a) somewhat helpful to me. (b) very helpful to me.	40	28	3.6
When I am analysing a story or a novel: (a) I think of the incidents and try to put them together to figure out the themes. (b) I just know what the themes are when, I finish reading and then I have to go back and find the incidents that demonstrate them	16	29	3.65
When I solve math problems: (a) I usually work my way to the solutions one step at a time. (b) I often just see the solutions but then have to struggle to figure out the steps to get to them.	12	30	3.75
I learn: (a) at a fairly regular pace. If I study hard, I'll "get it." (b) in fits and starts.,I'll be totally confused and then suddenly it all "clicks	24	31	4.5
When writing a paper, I am more likely to: (a) work on (think about or write) the beginning of the paper and progress forward. (b) work on (think about or write) different parts of the paper and then order them.	32	32	4.65
Once I understand: (a) all the parts, I understand the whole thing. (b) the whole thing, I see how the parts fit	8	33	4.7

while learning in a teaching environment which does not support their preferences.

It is perceived that only learners with mild preferences would take all the 11 questions within a particular dimension in the questionnaire. This perception is conceived on the assumption that if 60% of a user's answers are in favour of a single preference (X or Y) within a particular dimension, then that user would likely be all right in an environment which favours that specific preference. Yet for some courses, this threshold of 60% could be revised as deemed appropriate during system validation in real-world settings.

The flowchart in Fig. 4.7, illustrates the variable i which ranges over all four dimensions of the Felder-Silverman model while the variable j relates to the current question which is being processed within a dimension, hence ranging over all eleven questions within a dimension. The number of answers to "a"

Table 4.4: Order of Questions in Information Understanding Dimension

Questions	FSLSM Question's Sequence	ULEARN Question's Sequence	Difficulty level
D4: Information understanding (Sequential/Global)			
When I think about what I did yesterday, I am most likely to get: a) a picture. b) words	3	34	3.15
When I meet people at a party, I am more likely to remember: a) what they looked like. b) what they said about themselves	35	35	3.2
When I see a diagram or sketch in class, I am most likely to remember: a) the picture. b) what the instructor said about it	27	36	3.3
I prefer to get new information in: a) pictures, diagrams, graphs, or maps. b) written directions or verbal information.	7	37	3.5
When I get directions to a new place, I prefer: a) a map. b) written instructions	23	38	3.6
In a book with lots of pictures and charts, I am likely to: a) look over the pictures and charts carefully. b) focus on the written text.	11	39	3.7
I remember best a) what I see. b) what I hear.	19	40	3.75
I tend to picture places I have been: a) easily and fairly accurately. b) with difficulty and without much detail	43	41	4.5
I like teachers: a) who put a lot of diagrams on the board. b) who spend a lot of time explaining.	15	42	4.65
When someone is showing me data, I prefer: a) charts or graphs. b) text summarising the results.	31	43	4.8
For entertainment, I would rather: a) watch television. b) read a book	39	44	4.85

and "b" in the specific dimension i is computed in the variable A_i (resp. B_i), for $1 \leq i \leq 4$. In this way, once all four dimensions have been processed, the algorithm is deemed to be successfully completed for each student (for each single-case).

The following section presents a concise overview with respect to the working of the proposed algorithm by making use of a real academic example.

4.5 Modelling a real academic example

In this section, the working of the proposed algorithm for initialising students' learning styles will be tested through the modelling of a real academic case study.

Step 1—Fill-out FSLSM-ILS dynamic questionnaire.

The process begins when Clara, Emma, Bob and Oliver participate in the learning activity by interacting with ULEARN. Firstly, the system is expected to invite these students to go through the

```
Algorithm Initial_LS
Input: the ILS questionnaire structured
       as in Table 4.1
Output: an array LS[1..8] of learning styles
Begin
/* i ranges over the 4 dimensions */
/* j ranges over the 11 questions in i */
/* LS[2i-1] and LS[2i] correspond to the
    learning styles in the dimension i. */
 for i = 1 to 4 do
   A = 0;
   B = 0;
   j = 1;
   while (A<7 and B<7 and j<=11) do
       read answer to question j
                of dimension i;
       if (answer is "a") then
           A = A + 1;
       else
           B = B + 1;
       fi
       j = j+1;
   od
   LS[2i-1] = A/(A+B);
   LS[2i] = B/(A+B);
 od
End
```

Figure 4.6: Pseudo-code of the algorithm for calculating initial learning style

LS dynamic questionnaire while simultaneously filling it out so as to identify their initial learning styles. As each student goes through the questions, this dynamic questionnaire presents a reduced number of questions based on each of their answers. Table 4.5 illustrates an example of the number of questions asked for four different students within the D1 *(Active/Reflective)*. The best-case scenario in which the smallest number (28) of questions are taken is illustrated in the case where the student Clara, in the first dimension, selects the same answer to the first seven questions. On the other hand, students Emma and Bob are regarded as the worst-case scenarios where all the 11 questions have to be considered for a decision to be made. It is worth noting that the above-described scenario is the only situation where this happens, i.e., where all 11 questions are taken into consideration. Both Emma and Bob possess mild Learning Style preferences (refer to Section

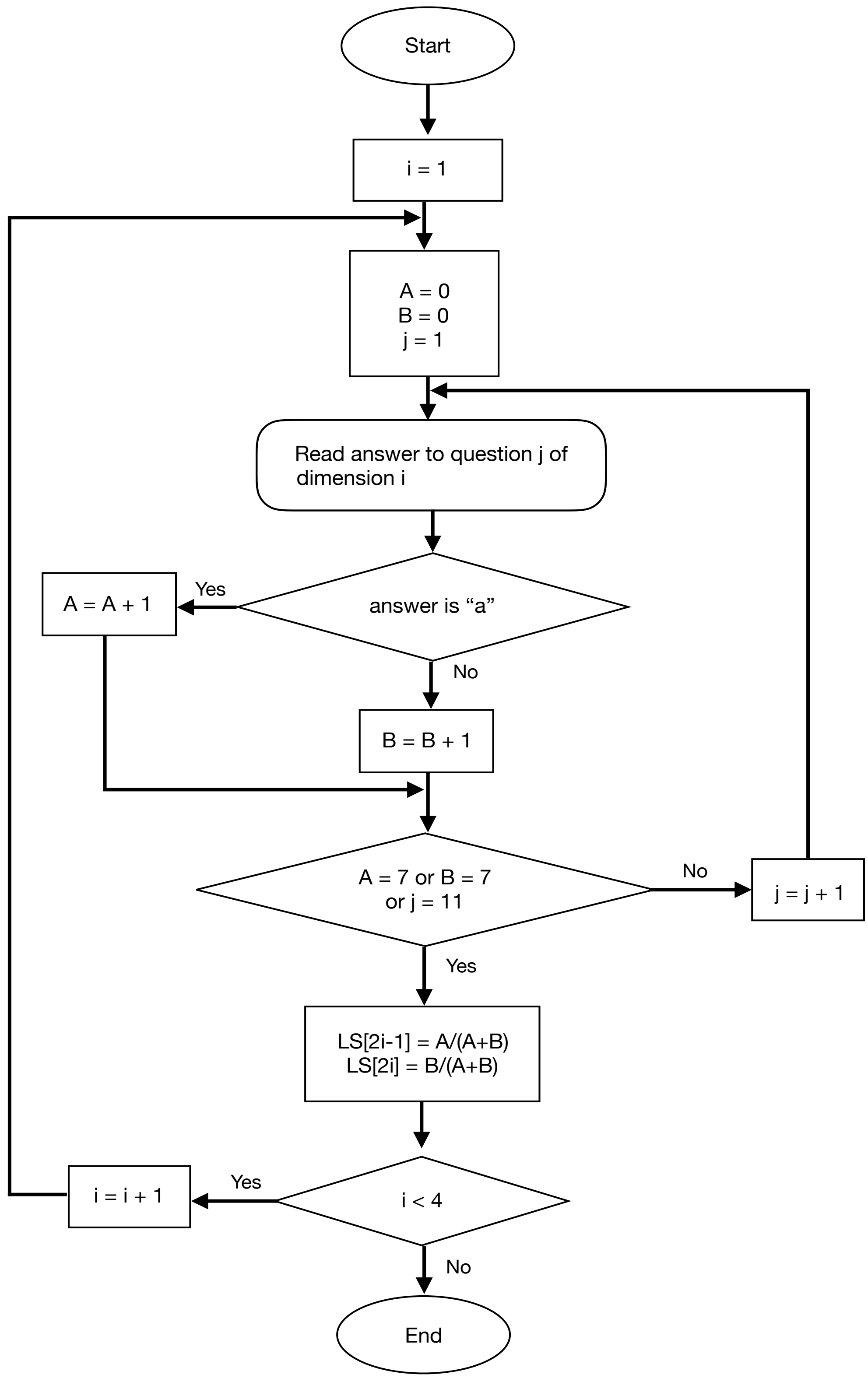


Figure 4.7: Flowchart of the algorithm for calculating the initial learning style of one student

Table 4.5: Examples of execution output Information Processing dimension of the algorithm Initial_LS

<table>
<tr><td></td><td colspan="8">Participants</td></tr>
<tr><td></td><td colspan="2">Clara</td><td colspan="2">Emma</td><td colspan="2">Bob</td><td colspan="2">Oliver</td></tr>
<tr><td>D1 (Information Processing)</td><td>Active</td><td>Reflective</td><td>Active</td><td>Reflective</td><td>Active</td><td>Reflective</td><td>Active</td><td>Reflective</td></tr>
<tr><td>Questions ULEARN Sequence</td><td>a</td><td>b</td><td>a</td><td>b</td><td>a</td><td>b</td><td>a</td><td>b</td></tr>
<tr><td>Q1</td><td>✓</td><td></td><td>✓</td><td></td><td></td><td>✓</td><td>✓</td><td></td></tr>
<tr><td>Q2</td><td>✓</td><td></td><td></td><td>✓</td><td></td><td>✓</td><td></td><td>✓</td></tr>
<tr><td>Q3</td><td>✓</td><td></td><td>✓</td><td></td><td>✓</td><td></td><td></td><td>✓</td></tr>
<tr><td>Q4</td><td>✓</td><td></td><td></td><td>✓</td><td>✓</td><td></td><td></td><td>✓</td></tr>
<tr><td>Q5</td><td>✓</td><td></td><td>✓</td><td></td><td></td><td>✓</td><td></td><td>✓</td></tr>
<tr><td>Q6</td><td>✓</td><td></td><td>✓</td><td></td><td>✓</td><td></td><td></td><td>✓</td></tr>
<tr><td>Q7</td><td>✓</td><td></td><td></td><td>✓</td><td></td><td>✓</td><td></td><td>✓</td></tr>
<tr><td>Q8</td><td colspan="2" rowspan="4">Skipped the rest of questions within D1</td><td>✓</td><td></td><td></td><td>✓</td><td>✓</td><td></td></tr>
<tr><td>Q9</td><td></td><td>✓</td><td>✓</td><td></td><td></td><td>✓</td></tr>
<tr><td>Q10</td><td>✓</td><td></td><td>✓</td><td></td><td colspan="2" rowspan="2">Skipped the rest of questions within D1</td></tr>
<tr><td>Q11</td><td></td><td>✓</td><td></td><td>✓</td></tr>
<tr><td rowspan="2">Total</td><td>#a</td><td>#b</td><td>#a</td><td>#b</td><td>#a</td><td>#b</td><td>#a</td><td>#b</td></tr>
<tr><td>7</td><td>0</td><td>6</td><td>5</td><td>5</td><td>6</td><td>2</td><td>7</td></tr>
</table>

4.2) within every dimension and are hence well-balanced individuals who can learn in any type of teaching environment.

Table 4.6 shows the total answers for four students in the four dimensions of the FSLSM-ILS questionnaire (as explained in Section 2.2.5). It is important to note that the notations #a and #b denote the number of answers pertaining to “a” and “b”, respectively.

Step 2–Calculating students’ initial learning styles.

In this step, the algorithm will perform the calculation of Learning Styles for (Clara, Emma, Bob, and Oliver) to initialise their Learning Styles based on their answers in four dimensions, as shown

Table 4.6: Examples of execution output (D1, D2, D3 and D4) of the algorithm Initial_LS

	Learning Style Dimensions								
	Processing		**Perception**		**Input**		**Understanding**		
	Active	**Reflective**	**Sensory**	**Intuitive**	**Visual**	**Verbal**	**Sequential**	**Global**	
	#a	**#b**	**#a**	**#b**	**#a**	**#b**	**#a**	**#b**	Total
Clara	7	0	7	0	0	7	0	7	28
Emma	6	5	6	5	5	6	6	5	44
Bob	5	6	4	7	4	7	7	4	44
Oliver	2	7	0	7	7	0	0	7	30

in Table 4.6. Every two criteria values within the same dimension are normalised to ensure they lie between 0 and 1.

For example, Emma's Learning Styles in D2 Information Perception (Sensory/Intuitive). As shown in Table 4.3, Emma's answers for Sensory = 6, whereas, Intuitive = 5 with the total number of answers for D2 (Information Perception) = 11, then Emma's Learning Styles, Sensory = 6/11 = 0.55, Intuitive = 5/11 = 0.45.

Table 4.7 presents the Learning Styles of four students computed according to their answers to the dynamic ILS questionnaire.

Table 4.7: Calculating Initial_LS

	Learning Style Dimensions							
	Processing		**Perception**		**Input**		**Understanding**	
	Active	**Reflective**	**Sensory**	**Intuitive**	**Visual**	**Verbal**	**Sequential**	**Global**
Clara	**7**	**0**	**7**	**0**	**0**	**7**	**0**	**7**
Clara's LSs	**7/7 = 1**	**0/7 = 0**	**7/7 = 1**	**0/7 = 0**	**0/7 = 0**	**7/7 = 1**	**0/7 = 0**	**7/7 = 1**
Emma	**6**	**5**	**6**	**5**	**5**	**6**	**6**	**5**
Emma's LSs	**6/11 = 0.55**	**5/11 = 0.45**	**6/11 = 0.55**	**5/11 = 0.45**	**5/11 = 0.45**	**6/11 = 0.55**	**6/11 = 0.55**	**5/11 = 0.45**
Bob	**5**	**6**	**4**	**7**	**4**	**7**	**7**	**4**
Bob's LSs	**5/11 = 0.45**	**6/11 = 0.55**	**4/11 = 0.36**	**7/11 = 0.64**	**4/11 = 0.36**	**7/11 = 0.64**	**7/11 = 0.64**	**4/11 = 0.36**
Oliver	**2**	**7**	**0**	**7**	**7**	**0**	**0**	**7**
Oliver's LSs	**2/9 = 0.22**	**7/9 = 0.78**	**0/7 = 0**	**7/7 = 1**	**7/7 = 1**	**0/7 = 0**	**0/7 = 0**	**7/7 = 1**

Step 3–Store Learning Styles values within a student's profile.

Finally, the proposed algorithm stores the student's initial Learning Styles in their profiles, as shown in Table 4.8.

Table 4.8: Student's initial learning styles

	Initial Student's profile(learning styles)							
Student	**Active**	**Reflective**	**Visual**	**Verbal**	**Sequential**	**Global**	**Sensing**	**Intuitive**
Clara	1	0	1	0	0	1	0	1
Emma	0.55	0.45	0.55	0.45	0.45	0.55	0.55	0.45
Bob	0.45	0.55	0.36	0.64	0.36	0.64	0.64	0.36
Oliver	0.22	0.78	0	1	1	0	0	1

Based on the initial profile (Learning styles), the algorithm will recommend the most suitable learning objects that match student's learning styles, as discussed in the next chapter.

4.6 Summary

This chapter focused on the proposal of a novel algorithm for initialising student's Learning Styles in line with a dynamic ILS questionnaire during the registration process. Through this approach, students are required to answer fewer questions within each dimension compared to the Felder-Silverman model for the purpose of determining their initial Learning Styles. Moreover, an empirical study was conducted for determining the sequence of questions within each of the four dimensions within the FSLSM questionnaire, which was then made use of to construct the dynamic algorithm that computes the user's initial Learning Style. As mentioned earlier, through this algorithm, initial Learning Styles can be inferred through fewer questions than the FSLSM model questionnaire; thus, saving a considerable amount of time and effort required to answer the 44 questions (eleven per dimension) within the FSLSM questionnaire. In addition to overcoming the issue related to a "cold-start", the proposed algorithm identifies students' Learning Styles correctly in order to improve recommendation accuracy. Furthermore, student profiles are used to make better-personalised recommendation approaches in Chapter 5.

Chapter 5

Learning Objects Recommendation

As explored in previous chapters, learning institutions are increasingly using e-learning recommender systems to enable and deliver learning anywhere and anytime to students. Yet the delivery of personalised learning objects on the basis of students' preferences continues to remain a challenge. Current mainstream recommendation algorithms, including Collaborative, Content-Based and Hybrid Filtering approaches are concerned with just two entities, namely users and items with their ratings [305, 306]. Nevertheless, the above-mentioned techniques fail to attend to students' learning styles and preferences, both of which are extremely crucial for an accurate learning process. Furthermore, there exists the issues of cold-start and data sparsity which is experienced by majority of existing recommendation approaches. The cold-start problem comprises the issue where no information is available early on upon which to base recommendations (e.g new student or new learning objects) on. Whereas, data sparsity problems are caused due to an insufficient number of the transactions and feedback data such as ratings.

Nevertheless, these methods do not pay attention to student's preferences, such as learning styles, which are especially important for the accuracy of learning objects prediction or recommendation. Moreover, several recommendation techniques experience cold-start and data sparsity problems. The cold-start problem where no information is available early on upon which to base recommendations (e.g new student or new learning objects). Whereas, data sparsity problem is caused by the insufficient number of the transactions and feedback data such as rating.

In this chapter, we specifically address these challenges by proposing a novel recommender algorithm so as to improve the quality of recommender systems. This technique would aid in recommending the top-n Learning Objects for students based on their individual learning styles.

This chapter addresses research questions 3 and 4:

> **RQ3.** *What similarity metrics should be used in the learning object recommendation algorithm to achieve the best accuracy?*
>
> **RQ4.** *How can personalised learning objects be recommended based on student's learning styles and object profiles?*

This chapter has a two-fold purpose:

- First, an empirical study has been conducted to devise an efficient similarity metric to use in the proposed recommend algorithm. In addition, an understanding on how changing similarity measures could further improve the recommendation process is developed through this study.
- Second, three novel recommendation algorithms (ECF, ECBF, and EHF) are proposed for recommending suitable learning objects on the basis of student learning styles. We adopt a novel mechanism to combine students' actual ratings with their individual learning styles and hence, solve issues of cold-start and data sparsity. So as to find the best approach for recommendation purposes, three recommendations algorithms have been tested, and their accuracy has been measured through the use of traditional evaluation metrics, including the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE). It is revealed through the results that the EHF algorithm possesses a higher accuracy compared to the ECF and ECBF recommendation methodologies (Section 5.2.4).

The rest of this chapter is organised as described. While a comparison of similarity metrics is presented in Section 5.1, Section 5.2 introduces the proposed learning objects recommendation module, whilst sub-section 5.2.4 covers the experimental results and analysis. Finally, Section 5.3 sums the entire chapter.

5.1 Comparison of different similarity matrices

Carrying out an experimental study was deemed important to identify and investigate relevant similarity metrics for the purpose of improving the accuracy of recommendation approaches based on learning objects. After studying current literature, four common similarity measures were identified as part of this experiment, namely, Euclidean Distance [296], Pearson's Correlation Coefficient [300], Cosine Similarity [302] and Manhattan Distance [303], were implemented (see Section 2.7).

Before applying the Proposed algorithm explained in Section 5.1.1, the K-means clustering algorithm (see Chapter 2) is applied for clustering learning objects into different clusters according to their profiles (learning styles), so that similar LOs are put together within the same cluster while dissimilar LOs are placed within different clusters. This goal of clustering LOs comprises reducing sparsity problems and increasing the algorithm performance while selecting appropriate similarity metrics since the data to be analysed is quite small [307]. The following section presents the algorithm for selecting similarity metrics to evaluate their performance within the context of e-learning.

5.1.1 Proposed algorithm for similarity metrics selection in the Context of e-learning

We have proposed *algorithm* 1 which considers two abstract similarity metrics Sim_1 and Sim_2 for use as place holders is described in five steps. Actual similarity metrics as presented in Section 2.7 of Chapter 2 replace these abstract similarity metrics to determine which combination produces the best accuracy in predicting top-n learning objects of students' ratings. The best performing similarity combination is retained later for the proposed recommender algorithm.

Thereby, a crucial question constitutes "what are the similarity metrics that provide the best prediction of the student ratings of the learning objects?", the answer to which will be determined in the following section through means of the experimental study. As mentioned previously, the goal of these experiments is to select similarity metrics with the highest accuracy to use them later in proposed algorithms for recommending personalised learning objects.

Algorithm 1 Proposed algorithm for similarity metrics selection

Let LS be the learning style vector of the active student.

Step 1 *– Calculate C, the nearest learning object cluster to the active student learning style LS using the similarity metric* Sim_1. This is done by calculating the similarity degree between LS and the centroid of each cluster and choose the cluster that produces the highest similarity degree.

Step 2 *– Calculate the similarity degree between LS and each learning object in C using the similarity metric* Sim_2. For all $OP \in C$, calculate $Sim_2(LS, OP)$.

Step 3 *– Select the top-n learning objects most similar to LS.* The number of learning objects to be selected can be a chosen constant or determined using a similarity threshold.

Step 4 *– Predict the Student's ratings of the top-n learning objects.* A 5-level Likert scale is considered, with 1 be the lowest score and 5 the highest score. The learner ratings of the learning objects are predicted using Eq. (5.1).

$$\tilde{r}(LS, OP_i) = int(0.5 + Sim_2(LS, OP_i) \times 5), \quad 1 \leq i \leq n \tag{5.1}$$

where $\tilde{r}(LS, OP_i)$ is the predicted rating of the learning object (profile) OP_i by the active student LS and $int(x)$ denotes the closest integer to the real value x; e.g. $int(2.3) = 2$ and $int(2.5) = 3$.

Step 5 *– Rank learning objects in descending order of predicted ratings.* This constitutes the list of learning objects recommended to the active student.

5.1.2 Experiment 1: Which similarity metrics provide the best prediction accuracy?

An experiment was carried out to determine which similarity metric would deem to be the best for measuring the similarity between students or/and learning objects in the e-learning domain. Hence, this experiment focuses on four metrics: the Euclidean Distance, the Pearson Correlation Coefficient, the Cosine Similarity, and the Manhattan Distance (See Sect. 2.7). With the aim of finding the best values for Sim_1 and Sim_2, the proposed *algorithm* 1 was used as the basis of this study. A brief overview of the dataset, similarity performance and results are covered in further sub-sections.

In the following sub-sections, we describe the dataset, evaluation of similarity performance, and the results and discussion.

5.1.2.1 Dataset

The dataset of the MOODLE log-file obtained from the School of Business located in AAST, Egypt particularly in the fall and spring semesters during the academic years 2016/2017 and 2017/2018 was used as part of this study. The experiment which was conducted on a total of 30 students in a specific course made use of the dynamic ILS questionnaire (Chapter 4) for the identification of students' learning styles. Networks and e-commerce was the course of interest which comprised 20 topics. Each topic comprised a minimum of 15 learning objects in various presentation styles. Through the course of the study, students were tasked with making use of a 5-level Likert scale (Table 5.1) to rate every learning object with 1 denoting "not at all useful" and 5 indicating "very useful".

Table 5.1: Rating scale

Linguistic term	Rating
Not at all Useful (NU)	(1)
Poor (P)	(2)
Fair (F)	(3)
Good (G)	(4)
very useful (VU)	(5)

By making use of Visual Studio in addition to Windows Presentation Foundation (WPF), the proposed algorithm was implemented in C++ for designing a GUI (Graphical User Interface), with the SQL server including learning styles, student ratings and learning object profiles. It is important to note that all tests covered in this chapter were run on a Windows-based PC comprising an Intel Core i5 processor running at a speed of 2.40 GHz as well as a RAM memory of 16 GB. Illustrated in Fig. 5.1 is the GUI which permits the selection of numerous similarity metric combinations.

5.1.2.2 Evaluation of similarity performance

In order to observe how accurately various similarity metrics make predictions, the Mean Absolute Error (MAE) (defined by Eq. (5.2)) [308, 309] and the Root Mean Squared Error (RMSE) (defined

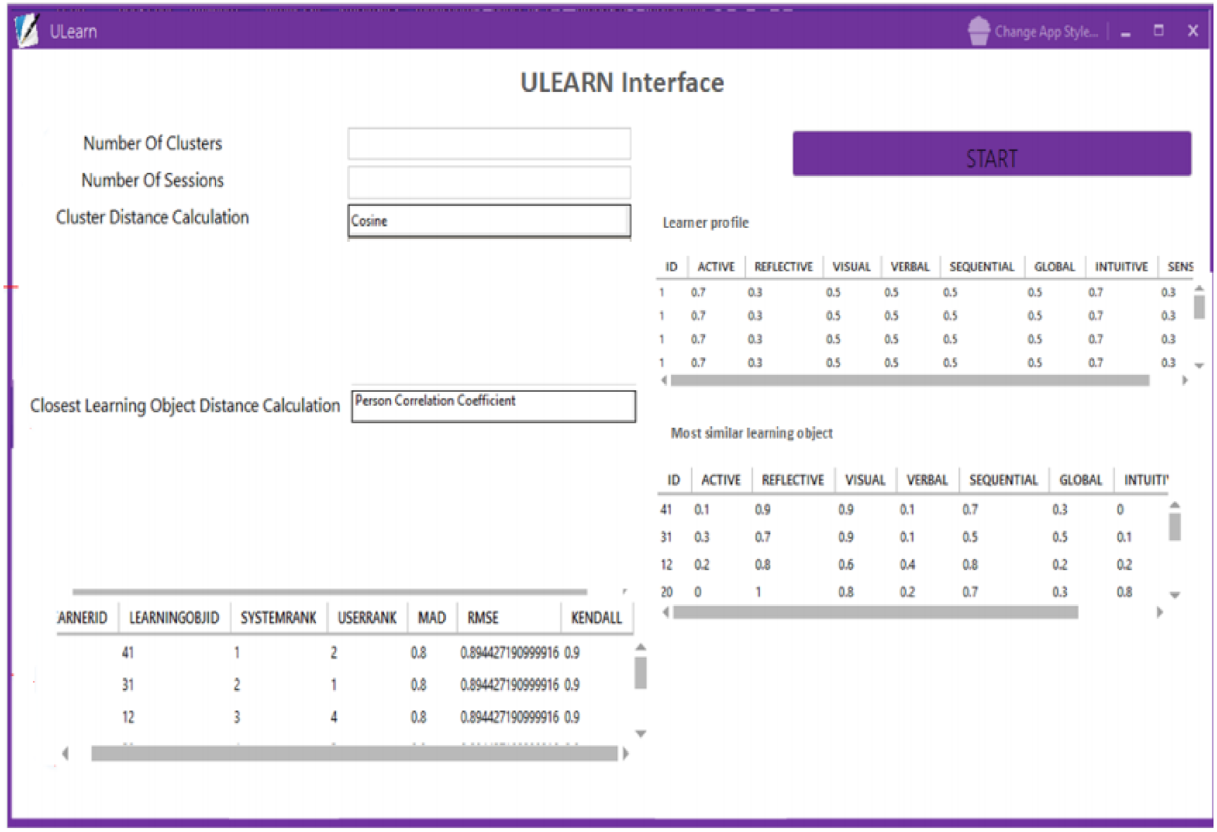


Figure 5.1: ULearn interface

by Eq. (5.3)) [310] are the most commonly used metrics, both of which aid in the measurement of the average magnitude of errors within a set of predictions and do not consider the directions of the ratings. While RMSE is more convenient in detecting large errors within a prediction, these may be observed very rarely with MAE. The values of the results range from 0 to ∞ where a smaller values denotes a higher accuracy. In the following equations, r_i denotes the actual student rating of the learning object i while $\tilde{r}_i$ indicates the predicted student rating for that learning object, $1 \leq i \leq n$.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |r_i - \tilde{r}_i| \tag{5.2}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (r_i - \tilde{r}_i)^2} \tag{5.3}$$

5.1.2.3 Results and Discussion

Numerous similarity metrics are evaluated and discussed in this section. Following the evaluation of the similarity metric Sim_1, it was inferred that the preciseness of the recommender algorithm was not affected. This similarity metric is made use of to compute the nearest learning object cluster to the active student's learning style.

Fig. 5.2 illustrates the distributions of a random sample of 30 students across K=3 by making use of the Euclidean (Eq. (2.1)), Manhattan (Eq. (2.6)), Person correlation coefficient (Eq. (2.4)), and cosine (Eq. (2.5)) similarity metrics respectively. From these results, it is inferred that the distributions do not differ from each other significantly, meaning that any one of them could be made use of for Sim_1 for the proposed algorithm in Section 5.1.1.

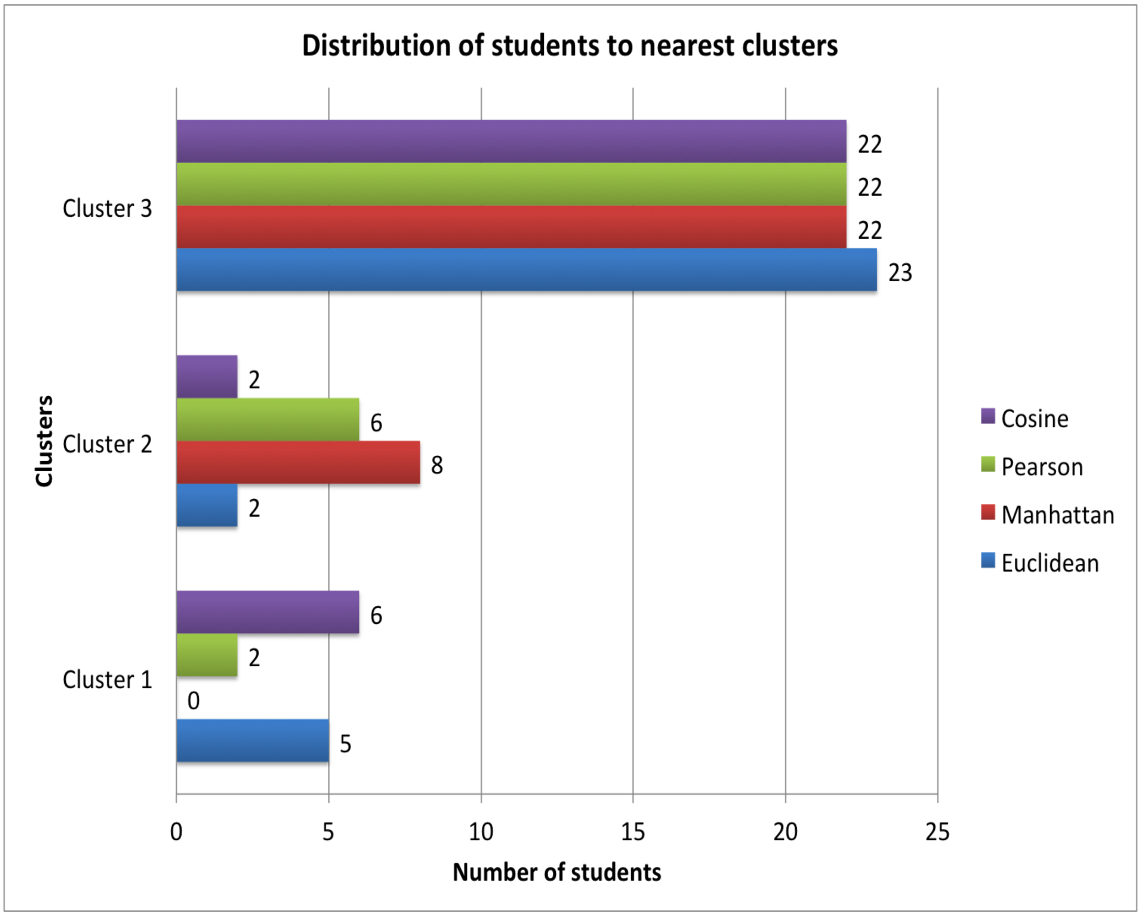


Figure 5.2: Distribution of students to nearest clusters

Since it is evident regarding which metric (Sim_1) to use for calculating the closest learning object cluster to a student learning style, the following step involves choosing an efficient metric (Sim_2) for selecting the top-n closest learning objects to the student learning style in a learning objects cluster.

Experimental results indicate the development of an extremely accurate algorithm when the Pearson correlation coefficient (Eq. (5.5)) is substituted as Sim_2, as illustrated in Fig. 5.3 and Fig. 5.4

based on metrics MAE and RMSE respectively.

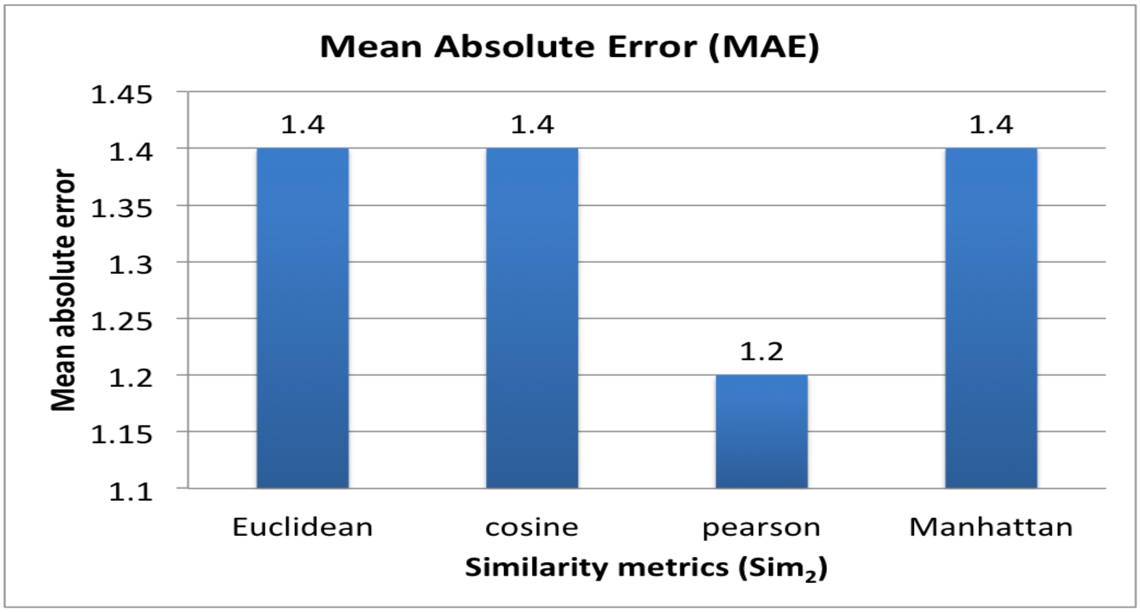


Figure 5.3: Accuracy of the recommender algorithm using MAE

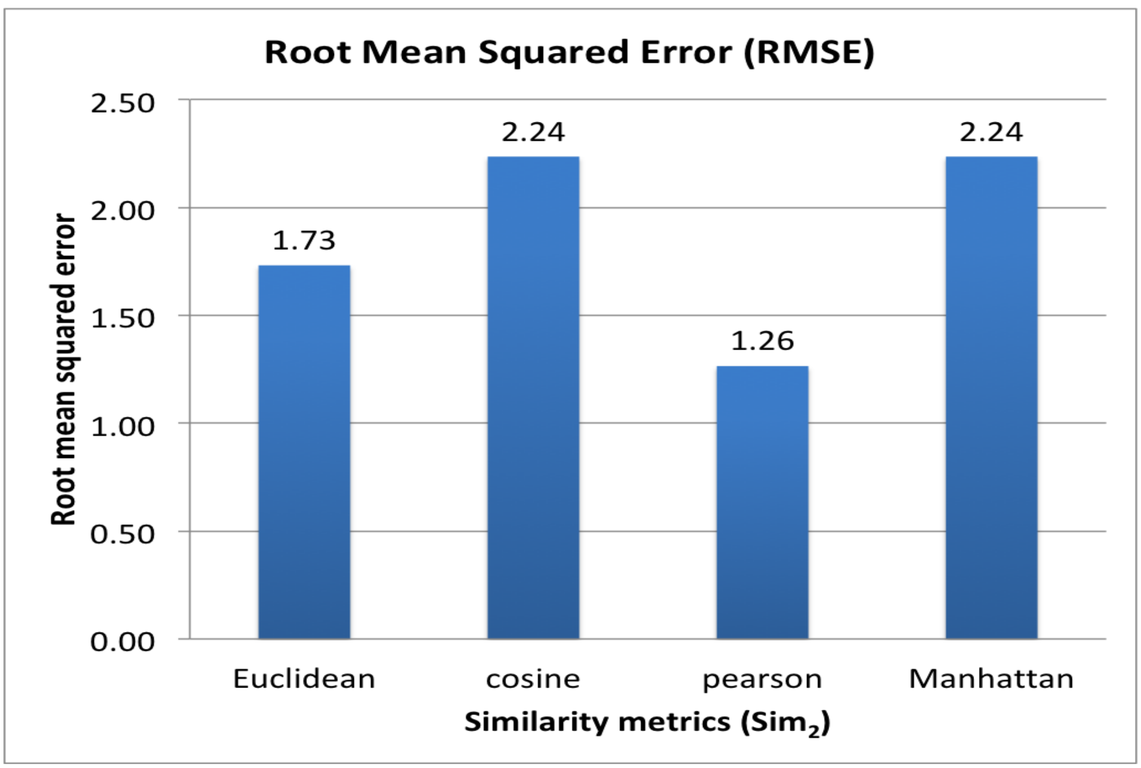


Figure 5.4: Accuracy of the recommender algorithm using RMSE

Thereby, following the conclusion of the study, it was inferred that *algorithm* 2 denotes the final algorithm where the Cosine similarity is made use of for determining the closest learning object to the active student learning style.

This cosine metric ranging between 0 and 1 is a similarity measure between two vectors and is denoted by Eq. 5.4 where X represents the user and Y indicates the cluster.

$$c(x,y) = \frac{x.y}{||x||.||y||} \tag{5.4}$$

Moreover, as mentioned previously, the Pearson correlation coefficient is utilised to select the top-n learning objects of the nearest cluster Y that is most similar to the active student learning style X.

$$P(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{5.5}$$

It is understood that the function $quickSort(L,i)$ sorts a list of tuples L along the i^{th} dimension. In the case of Algorithm 2, L is a list of tuples of the form $< x,y >$, where x is a learning object profile (OP) and y the corresponding predicted rating for the active student LS.

In the following section, we will explain the proposed recommendation algorithms used in recommending learning objects according to students' learning styles.

5.2 Learning Objects recommendation Module

Functioning as a middle layer in the architecture of the proposed e-learning recommender system, the Learning Objects Recommendation Module is responsible for generating and delivering top-n learning objects to students through sessions based on their learning style stored in their profile. In this section, we propose three distinct algorithms (each of which are described in the following sub-sections) for recommending top-n learning objects, on the basis of three approaches: Enhanced Collaborative Filtering, Enhanced Content-Based Filtering and Enhanced Hybrid Filtering. When compared to traditional techniques, these approaches are well-known for their efficient handling of cold-start and data sparsity issues by using information from students' learning styles and learning objects profiles. Following this, an experimental study was undertaken for the purpose of selecting a well performing algorithm.

Algorithm 2 Final Algorithm

```
input  : K: Number of clusters, K ≥ 1
         C: Set of K clusters C_1, ..., C_k
         CC: Set of the K clusters' centroids
         LS: Active student learning style
         T: Similarity degree threshold
output: TopN: List of recommended learning objects
j = 1
   x = c(LS, CC_1)
   /* c is the cosine similarity                                    */
for i = 2 to K do
    if c(LS, CC_i) > x then
        x = c(LS, CC_i)
          j = i
    end
end
/* C_j is the nearest cluster to LS                                 */
TopN = [] /* empty list                                             */
foreach OP ∈ C_j do
    S = P(LS, OP)
       /* P is the Pearson similarity                               */
    if S ≥ T then
        TopN.append(< OP, int(0.5 + S × 5) >)

    end
end
quickSort(TopN, 2)
 return TopN
```

5.2.1 Enhanced Collaborative Filtering recommendation algorithm

As previously explained in Chapter 2, traditional approaches of Collaborative Filtering rely heavily on co-rated items, meaning that performance of the recommender system will be lower in cases where matrices are generated from noisy ratings provided by students. Yet, similarity cannot be calculated when there exist no rated items, also known as the issue of cold-start. Thus, using our proposed novel model, the situation can be resolved to avoid any problems and significantly enhance the accuracy and quality of recommendation, as presented in algorithm 3. The proposed ECF is implemented in the following way: Let *LS* be the learning style vector of the active student.

Algorithm 3 Enhanced Collaborative-Filtering recommendation algorithm

1. Apply K-means to cluster the students profiles.
2. Select cs the nearest SP cluster to LS as in Eq. (5.4)
3. Foreach LO x
   (a) Let $I =$ set of the top-n nearest elements to LS in cs **that have rated** x as in Eq. (5.5)
   (b) If $\|I\| > 0$ then calculate the predicted rating for x as in Eq. (5.6)
   (c) If $\|I\| = 0$ then calculate the predicted rating for x as in Eq. (5.7)
4. Recommend the top-n highly rated LOs.

$$\tilde{r_1}(LS,x) = \frac{\sum_{u\in I} P(LS,u) \times r(u,x)}{\sum_{u\in I} P(LS,u)} \tag{5.6}$$

Where $\tilde{r}(LS,x)$ denotes the predicted rating value of the LO x for the active student LS. $P(LS,u)$ donates the Pearson correlation coefficient (defined as in Eq. (5.5)) of the two vectors LS and u; and $r(u,x)$ represents the actual rating of the LO x by the student u.

$$\tilde{r_2}(LS,x) = int(0.5 + P(LS,x) \times 5) \tag{5.7}$$

Equation (5.7) is used to solution the cold-start and the rating sparsity problems (case $\|I\| = 0$). In this case the predicted rating is measured as the similarity between the LO x and the active student LS multiplied by the maximum rating value which is 5. The value 0.5 is added so that the result is and integer between 1 and 5.

5.2.2 Enhanced Content-Based Filtering recommendation algorithm

As discussed earlier in Chapter 2, a general principle of CBF techniques comprises the identification of common characteristics for learning objects which receive a favourable rating from the learner followed by recommending to their new LO that shares these characteristics. An algorithm of a similarity model has been proposed in this thesis so as to enhance the accuracy of recommendation. The similarity between *SP* and *LO* is calculated using the algorithm 4:

Let *LS* be the learning style vector of the active student.

Algorithm 4 Enhanced Content-Based Filtering recommendation algorithm

(a) Let $\mathscr{O}$ be the set of all learning objects rated by LS.

(b) If $\mathscr{O} \neq \emptyset$ then

   i. Apply K-means to cluster $\mathscr{O}$

   ii. Foreach LO x

      A. Let $co_x =$ the nearest LO cluster to x as in Eq. (5.5)

      B. Let $J =$ set of the top-n nearest elements to x in co_xas in Eq.(2.5)

      C. Calculate the predicted rating for x as in Eq. (5.8)

   iii. Recommend the top-n highly rated LOs.

(c) If $\mathscr{O} = \emptyset$ then

   i. Apply K-means to cluster all the learning learning objects

   ii. Let $co =$ the nearest LO cluster to LS

   iii. Foreach $x \in co$

      A. Calculate the predicted rating for x as in Eq. (5.7)

   iv. Recommend the top-n highly rated LOs in co.

$$\tilde{r}_3(LS,x) = \frac{\sum_{u \in J} P(x,u) \times r(LS,u)}{\sum_{u \in J} P(x,u)} \tag{5.8}$$

5.2.3 Enhanced Hybrid Filtering Filtering recommendation algorithm

Since both approaches of CF as well as CBF possess certain benefits and limitations, the resulting accuracy of the recommender system will not be very high when making use of only one recommended algorithm. Hence, this has led scholars to propose a means of integrated recommendation algorithms to be used during the process of recommending [248, 249].

For enhancing the accuracy and quality of recommendation, our proposed EHF is implemented as shown in algorithm 5:

Let LS be the learning style vector of the active student.

Algorithm 5 Enhanced Hybrid Filtering recommendation algorithm

1. Let α be the weight of CF in the hybrid model; $0 \leq \alpha \leq 1$.
2. Apply K-means to cluster the students profiles
3. Select cs the nearest SP cluster to LS
4. Let $\mathscr{O}$ be the set of all learning objects rated by LS.
5. Apply K-means to cluster $\mathscr{O}$
6. Foreach LO x
   (a) Let $I =$ set of the top-n nearest elements to LS in cs **that have rated** x
   (b) Let $co_x =$ the nearest LO cluster to x
   (c) Let $J =$ set of the top-n nearest elements to x in co_x
   (d) If $\|I\| > 0$ and $\|J\| > 0$ then calculate the predicted rating for x as in Eq. (5.9)
   (e) If $\|I\| = 0$ and $\|J\| > 0$ then calculate the predicted rating for x as in Eq. (5.8)
   (f) If $\|I\| > 0$ and $\|J\| = 0$ then calculate the predicted rating for x as in Eq. (5.6)
   (g) If $\|I\| = 0$ and $\|J\| = 0$ then calculate the predicted rating for x as in Eq. (5.7)
7. Recommend the top-n highly rated LOs.

$$\tilde{r}(LS,x) = \alpha \times \tilde{r}_1(LS,x) + (1-\alpha) \times \tilde{r}_3(LS,x) \tag{5.9}$$

Note that in Eq. (5.9), the value of α is between 0 and 1; and $\tilde{r}_1(LS,x)$ and $\tilde{r}_3(LS,x)$ are defined as in Eq. (5.6) and Eq. (5.8), respectively. Here are some examples:

Note the in Eq. (5.9), the value of α is between 0 and 1. Here are some examples:

– $\tilde{r}(LS,x) = 0.5 \times \tilde{r}_1(LS,x) + (1-0.5) \times \tilde{r}_3(LS,x)$

$= 0.5 \times \tilde{r}_1(LS,x) + 0.5 \times \tilde{r}_3(LS,x)$

$= \frac{\tilde{r}_1(LS,x) + \tilde{r}_3(LS,x)}{2}$

– $\tilde{r}(LS,x) = 0.2 \times \tilde{r}_1(LS,x) + (1-0.2) \times \tilde{r}_3(LS,x)$

$= 0.2 \times \tilde{r}_1(LS,x) + 0.8 \times \tilde{r}_3(LS,x)$

- $\tilde{r}(LS,x) = 0.8 \times \tilde{r}_1(LS,x) + (1-0.8) \times \tilde{r}_3(LS,x)$
$$= 0.8 \times \tilde{r}_1(LS,x) + 0.2 \times \tilde{r}_3(LS,x)$$

- $\tilde{r}(LS,x) = 0.75 \times \tilde{r}_1(LS,x) + (1-0.75) \times \tilde{r}_3(LS,x)$
$$= 0.75 \times \tilde{r}_1(LS,x) + 0.25 \times \tilde{r}_3(LS,x)$$

5.2.4 Experiment 2: Which recommendation algorithm has the highest recommendation accuracy?

In this section, we conduct experiments to infer appropriate recommendation techniques to be used later for recommending learning objects within proposed e-learning systems. Specifically, we aim to resolve the following questions:

> ”What is the recommendation algorithm that provides the most suitable top-n learning objects that match students learning styles ?”

With the purpose of evaluating and testing different aspects of the proposed method and proving that the proposed algorithm to predict ratings delivers the most accurate results, the student dataset (see Sect.5.1.2.1) was divided into various parts, including:

1. Cold-start students: a set of students with lower than 5 ratings
2. Cold-start learning objects: a set of new LOs
3. All students

A cold-start was used for the assessment of the ability of algorithms in predicting ratings for those students with a few learning objects, providing less information to students. As mentioned above, the main aim of this included the way in which additional sources of information, including students’ learning styles could be made use of in addition to rating information so as to enhance the accuracy of rating predictions.

5.2.4.1 Evaluation on Rating Prediction

Following the pre-process on the student dataset, fifteen students were chosen in conjunction with 15 randomly chosen learning objects, as shown in Table 5.2 along with the computed predicted ratings for the learning objects. The effectiveness of a recommender algorithm mostly refers to its prediction accuracy [311]. From the experimental results in the table, it can be inferred that the EHF-0.5 algorithm is the most accurate, while on the other hand, it is presumed that EHF-0.5 has the least value of MAE from Figure 5.5 and thereby, excels in providing better predictions. The MAE value of EHF-0.5 is 0.6, whilst that of ECBF is 1.52, which is the greatest with respect to other approaches. Thus, it is known that the latter method would generate the least accurate predictions. Moreover, experimentally and theoretically, it is already proven that RMSE is always greater than the MAE, as illustrated in Figure 5.5. Also, the proposed EHF-0.5 algorithm repeatedly delivers a smaller RMSE than the others, indicating to be the most accurate.

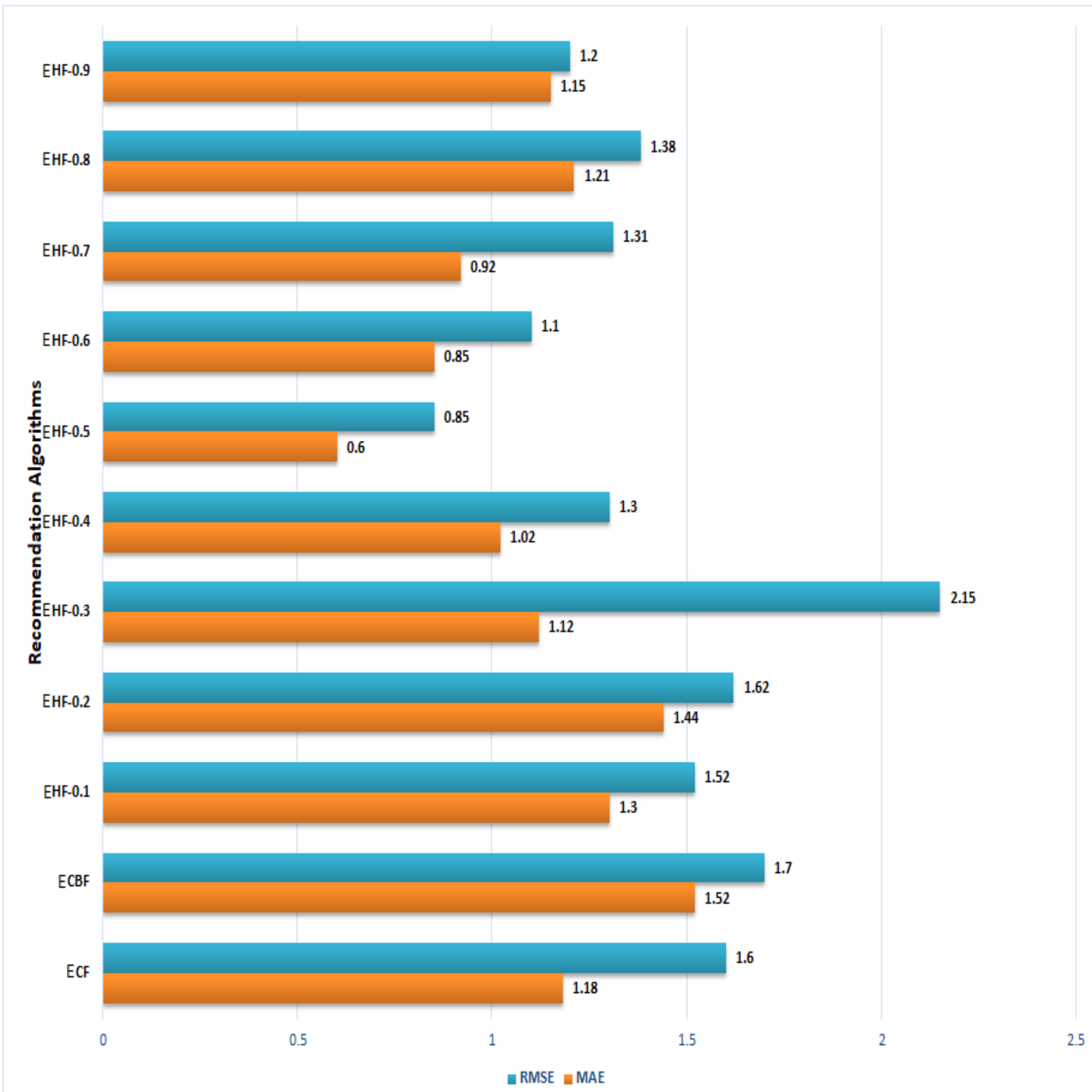


Figure 5.5: Accuracy of the recommender algorithm using MAE and RMSE

5.2.4.2 Evaluation of Cold-start

The experimental study was repeated from a different point of view for the purpose of evaluating the proposed approaches on the basis of handling the cold-start problem.

- **New students:** By incorporating personalised learning styles of students with their ratings, it is possible that the three different algorithms (ECF, ECBF and EHF-0.5) is capable of dealing with new students as shown in Figures 5.6 and 5.7 which compare the accuracy of various recommendation algorithms.

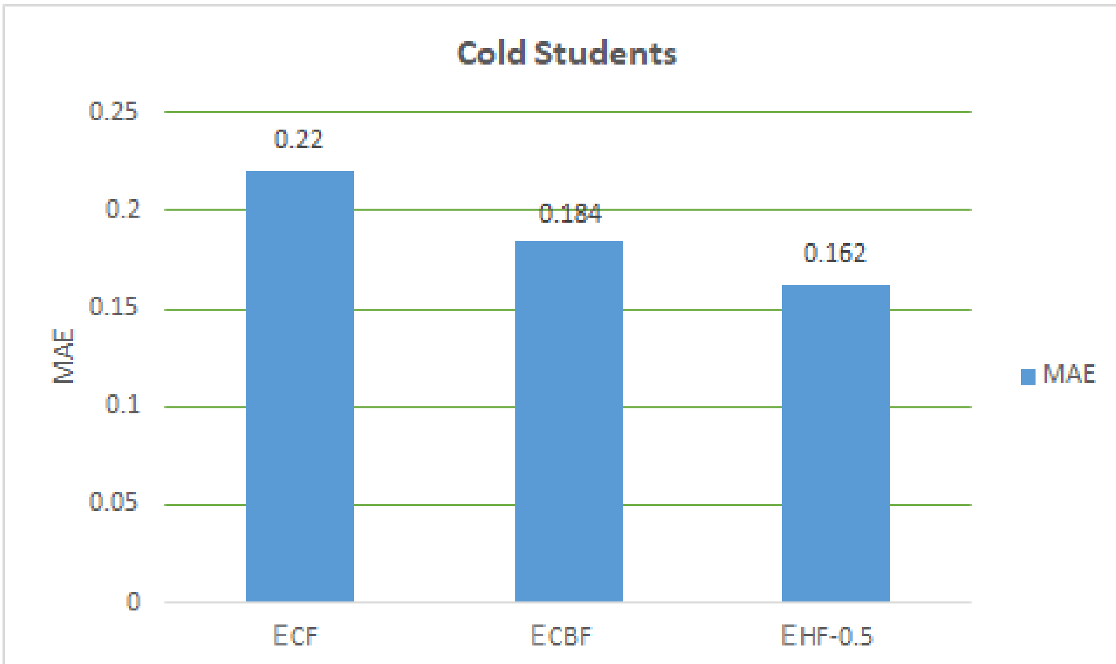


Figure 5.6: Performance comparison for cold students using MAE

- **New learning objects:** The three algorithms can make recommendation for new learning objects through measurement of the similarity between a learning object profile and a student's learning styles. From Figures 5.8 and 5.9, it is inferred that EHF-0.5 invariably outperforms in every experiment, thus indicating that our model handles new items better than ECF and ECBF.

- **New students and learning objects:** A special case is where neither the student nor the learning objects exist in the previous user-item rating matrix where a majority of current algorithms are not capable of dealing with such a situation. Yet, our proposed algorithm is able to provide recommendations based on relations between students and learning object

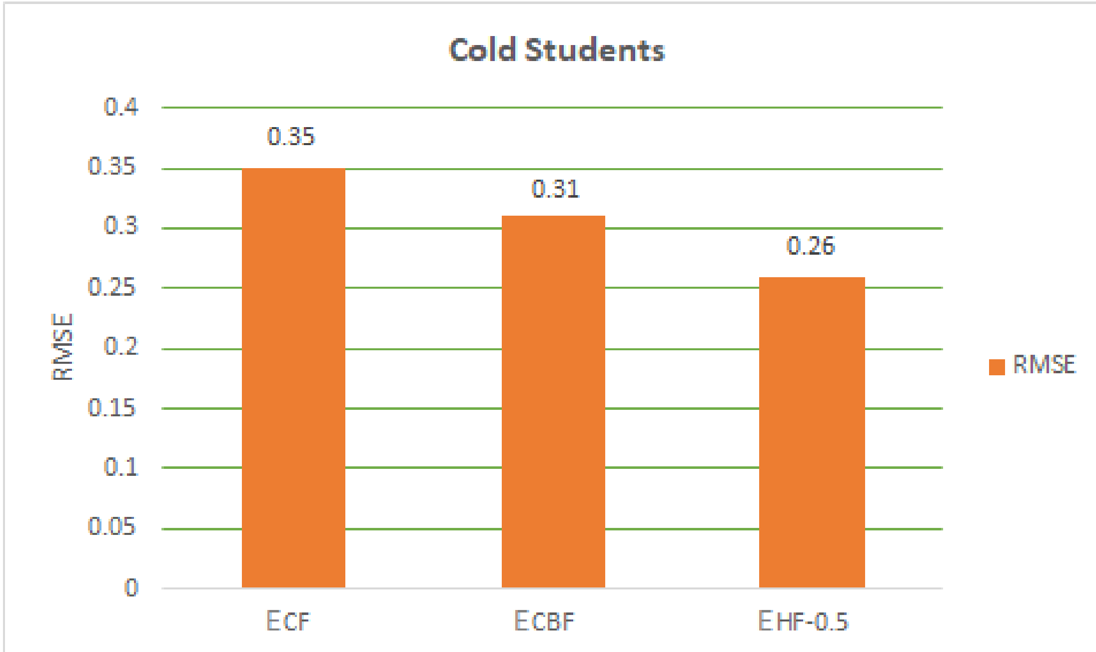


Figure 5.7: Performance comparison for cold students using RMSE

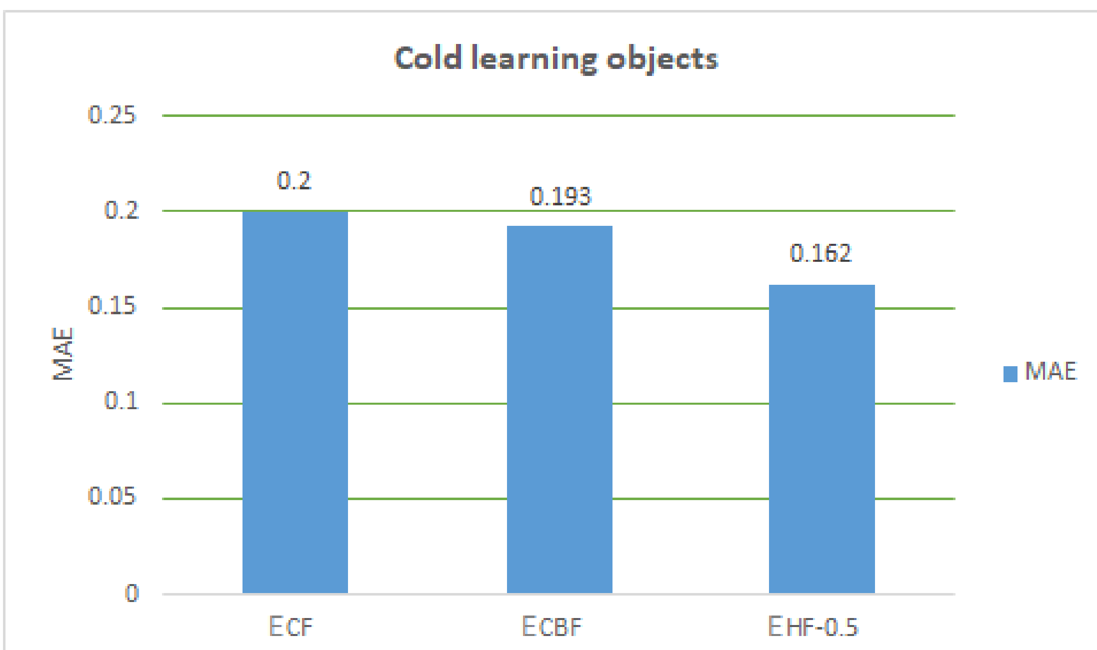


Figure 5.8: Performance comparison for cold LOs using MAE

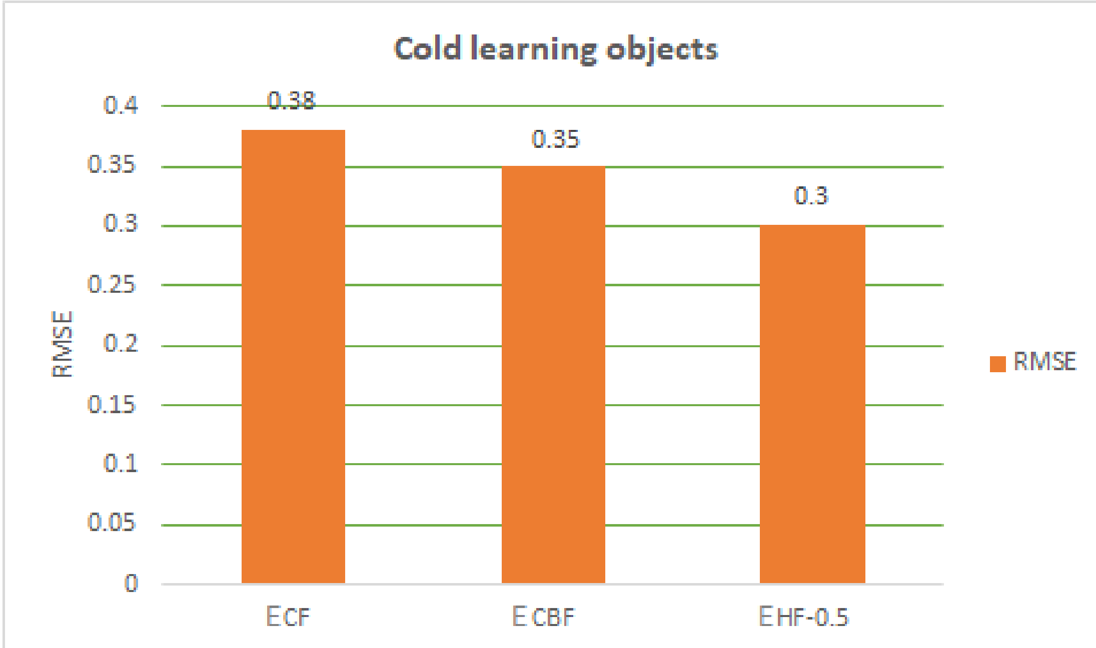


Figure 5.9: Performance comparison for cold LOs using RMSE

profiles.

It is also worthy to note that the results achieved by the hybrid filtering technique are impressive. Given the above results, analysis, and discussion following the experiment, it was concluded that the proposed algorithm EHF-0.5 performs better than ECF and ECBF. Furthermore, it was inferred that the EHF-0.5 hybrid algorithm handles problems related to cold-start and data sparsity efficiently as compared to the other approaches.

5.3 Summary

This chapter covers the recommendation algorithms in ULEARN which is designed for addressing problems of cold-start and data sparsity, particularly within the education domain. Basically divided into two stages so as to briefly discuss algorithms, the first stage includes a proposed algorithm to enable the selection of a best actual similarity metric from the pool commonly used similarity metrics. For this purpose, the K-means clustering algorithm was applied to produce K numbers of learning object clusters on the basis of students' profiles. Following this, an experimental study was conducted so as to determine the best actual similarity metrics used in e-learning recommender systems. The algorithm was found to perform best when the cosine similarity metric is utilised to compute the nearest learning object cluster and the Pearson correlation coefficient is made use of for selecting the top-n learning objects within the cluster.

The second stage presents an innovative algorithm for recommendation of the top-n learning objects in e-learning systems in line with students' learning styles, where we have concentrated on the significant improvement of the accuracy and quality of recommendations with respect to cold start and data sparsity issues. To propose an improved top-n learning objects recommender algorithm, we have tested and compared the performance of three approaches, namely, hybrid filtering, collaborative filtering, and content-based filtering.

After these experiments, it was inferred that the proposed EHF-0.5 hybrid algorithm produces the most accurate predictions. It was also ascertained that the EHF-0.5 hybrid algorithm effectively deals with issues related to a cold-start and rating sparsity.

In the next chapter, we will see how the student profile (learning styles) is automatically updated based on students' learning behaviour patterns using a novel algorithm for dynamic student profile adaptation.

Table 5.2: Showing predicted rating using the proposed algorithm

			Predicted rating using										
S.ID	LO.ID	Act. R	ECBF	ECF	EHF-0.1	EHF-0.2	EHF-0.3	EHF-0.4	EHF-0.5	EHF-0.6	EHF-0.7	EHF-0.8	EHF-0.9
1	1	2	1	2	1	2	2	2	2	3	3	2	1
2	5	4	3	3	4	4	4	4	4	3	4	4	4
3	13	2	2	3	2	3	3	3	2	2	3	3	3
4	25	5	2	3	2	5	5	5	5	4	5	5	3
5	10	2	3	3	3	2	1	3	2	3	2	2	3
6	1	3	1	2	2	2	3	3	3	2	3	3	2
7	5	2	1	2	1	2	2	2	2	2	2	2	2
8	13	4	3	4	4	4	4	4	4	4	4	4	2
9	15	4	2	2	2	3	3	4	4	4	4	3	2
10	20	5	3	4	3	4	3	4	5	4	4	4	3
11	22	2	1	2	2	2	2	2	2	2	2	2	3
12	4	3	1	2	3	3	3	3	3	3	3	3	3
13	8	5	3	4	3	4	4	4	5	4	4	3	3
14	7	4	3	3	3	3	4	4	4	4	4	3	3
15	16	5	2	4	2	3	5	5	4	5	4	4	3

S.ID= Student ID LO.ID= Learning Object ID Act.R= Actual Rating

Chapter 6

Dynamic Student Profile Adaptation

As discussed in the previous chapters, constructing an accurate and comprehensive student profile is one of the crucial issues in developing personalised learning environment. Current adaptive e-learning recommendation techniques experience a number of challenges that could restrain the personalisation features. One such critical drawback that these systems suffer from includes the way in which student preferences are modelled to be fixed over time. The main problem with such a modelling technique is that it ignores the fact that a student's preferences specifically learning styles are never fixed but they constantly modified over a period of time [147]. However, keeping an eye on a student's behaviour manually is a difficult task. Therefore, most of the adaptive e-learning systems which were examined earlier rely on static student information and cannot cope with the changes in their learning behaviour patterns, eventually leading to irrelevant course materials being recommended to students (see Chapter 2). For such systems to remain effective, it is necessary to create a student profile accurately and dynamically for better-personalised recommendations.

In this chapter, we first look at these problems of student profile adaptation and learning styles detection from their behaviour and attempt to answer the following research question:

> **RQ5.** *How to model dynamic student profiles based on student learning behaviour patterns?*

We propose a student profile adaptation which is able to dynamically capture, learn and adapt student's learning styles in a dynamic fashion implicitly. The first step includes designing a novel algorithm to monitor learning behaviour patterns of students in addition to capturing their learning styles and maintaining their student profiles confined in the Recommendation System (RS). Following this, the proposal of an innovative method to extract features which characterises learning behaviours and identifies learning styles on the basis of the FSLSM model is initiated.

This chapter is organised in the following way. The proposed algorithm's main concepts and architecture are covered in the Section 6.1. Next, Sub-section 6.2.1 describes the proposed algorithm in detail while sub-sections 6.2.2 and 6.2.3 presents relevant concepts related to student profile adaptation. Next, an example in order to provide more understanding to the reader has been demonstrated in Section 6.3 in addition to the experimental results and analyses which have been covered in Section 6.4. Lastly, Chapter 6.5 presents a brief summary of this chapter.

6.1 Proposed architecture for dynamic student Profiling

The architecture of the proposed algorithm in order to build and regularly update student profiles on the basis of observing students' behaviour patterns and learning objects is covered in this section. Following the recording a students' actions, the agent then makes use of these actions to construct the student profile. The proposed DSP Module's architecture is illustrated in Fig. 6.1. where it is divided into four main phases, namely, tracking learning behaviour patterns, dynamic checking sessions, student profile adaptation and updating student profiles.

– ***Phase 1: Tracking learning behaviour patterns.*** It is crucial that information about learning behaviour patterns are known so as to learn students' learning styles. Since the process of data collection poses an explicit additional burden on students [203], we aim at collecting student learning behaviour implicitly. Within this phase, the crucial task of analysing and extracting information related to student behaviour from the system log-file is performed. This includes time spent by the student on every learning object, the type of learning objects accessed (e.g. video, audio, or text), and the number of messages exchanged (refer to Section 6.2.1). Indeed, this phase is a very important part of student profile adaptation since capturing inaccurate student behaviour could directly affect the subsequent phases as well as the learning styles detection.

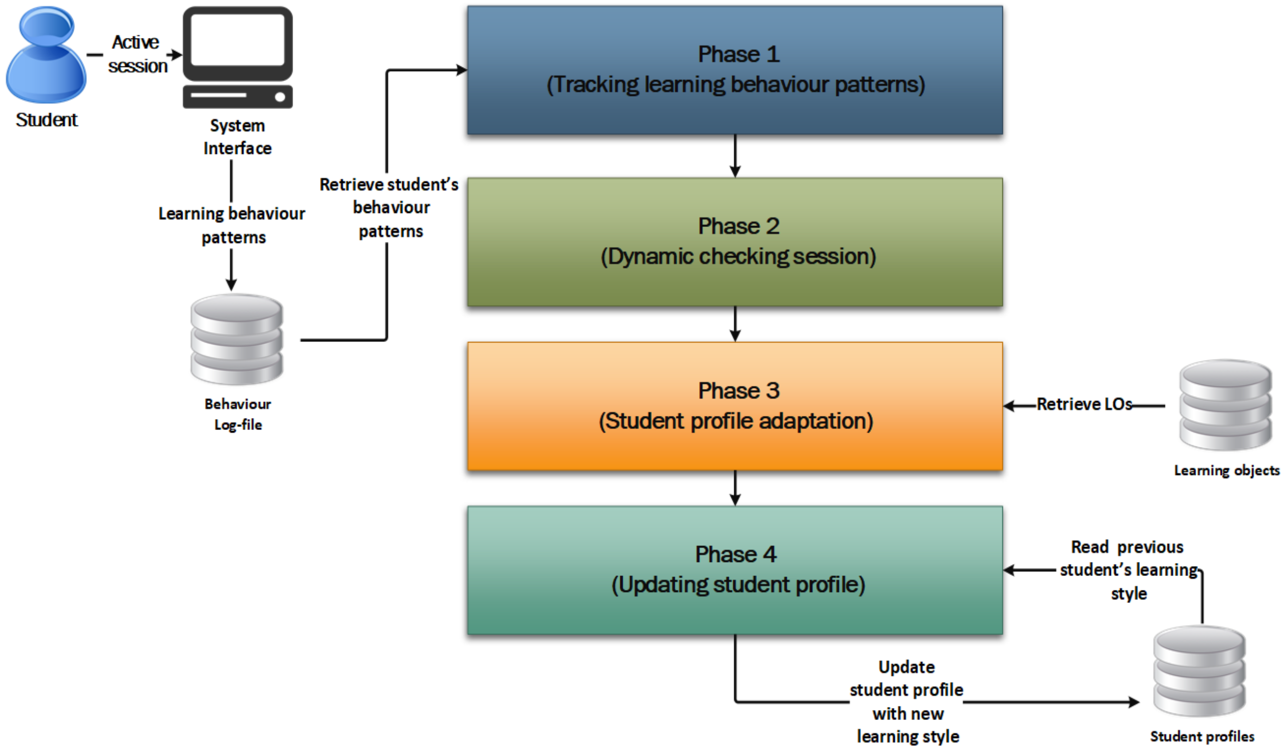


Figure 6.1: Architecture for updating student profile dynamically

– ***Phase 2: Dynamic checking sessions.*** So as to build student profiles, it is essential that necessary observation, processing and learning of student behaviour patterns takes place. Therefore, this step is liable for the effective management of a dynamic student profile adaptation process. *Dynamic checking sessions* send a request to *Student profile adaptation* to recalculate student's learning styles after the students perform several actions during the sessions since the last calculation of their learning styles.

– ***Phase 3: Student profile adaptation.*** Responsible for learning, modelling and adapting students' profiles, this phase has a significant role in our model. It is also in this phase that student behaviour patterns which were identified in *Phase 1* are assessed over a number of sessions after which corresponding feature vectors are aggregated for approximating current learning style preferences of students (refer to Section 6.2.2).

– ***Phase 4: Updating student profile.*** In this phase, the student's previous learning style and the current learning style (obtained in Step 3) are made use of to compute the student's new learning style. The student profile database is then used to store this data to be utilised for providing students with personalised learning objects the next time they log in (refer to Section 6.2.3).

The four phases of the proposed adaptation architecture in addition to the way in which learning styles are calculated have been discussed in depth in the following sections.

6.2 Proposed algorithm for dynamic student Profiling

In this section, a novel algorithm for building and frequently updating SP learning styles is proposed. We start by describing how to capture student learning behaviour patterns while s/he is taking a course via an e-learning system in Section 6.2.1. We then detail the student profile adaptation algorithm during sessions in Section 6.2.2. Lastly, in Section 6.2.3 we show how the student profile will be updated according to their learning behaviour patterns.

6.2.1 Capturing student learning behaviour patterns

As presented earlier in Chapter 2, students are even more burdened when data is collected explicitly and hence through this algorithm, we aim for the implicit collection of student learning behaviours. The critical objective of this phase includes addressing the question associated with which behaviour patterns should be selected to reflect the student's learning styles. So as to propose a general algorithm which could be applied to any learning system, it was crucial to make use of generic behaviour patterns which could be collected in any learning system so as to base approaches on. Three types of information were collected: time spent on each LO; number of messages; and the format of the LOs accessed during a specific session.

6.2.1.1 Time spent

The behaviour pattern for time spent W_i is determined as in Eq. (6.1), for every LO_i. Table 6.1 covers the rules related to the determination of learning styles corresponding to a given behaviour pattern.

Table 6.1: Behaviour pattern based on learning object type and FSLSM

Learning object	Behaviour pattern	Pattern description	Learning style criteria	
video	Time_spent	Total time spent on video content (Based on predefined video actual time)	>= 50% Active	< 50%Reflective
			>= 50% visual	< 50% verbal
Audio	Time_spent	Total time spent on Audio content (Based on predefined Audio actual time)	>= 50%Reflective	< 50%Active
			>= 50% verbal	< 50% visual
Simulation	Time_spent	Total time spent on simulation content (Based on predefined simulation actual time)	>= 50% Active	< 50%Reflective
PPT	Time _spent	Total time spent on PPT content (Based on total session duration)	>= 50%Global	< 50%Sequential
			>= 50% Active	< 50%Reflective
			>= 50% Intuitive	< 50%Sensing
PDF and Doc	Time_spent	Total time spent on PDF content (Based on total session duration)	>= 50%Sequential	< 50%Global
			>= 50%Reflective	< 50%Active
			>= 50% Intuitive	< 50%Sensing
			>= 50% verbal	< 50% visual
Summary	Time_spent	Total time spent on Summary content (Based on total session duration)	>= 50%Global	< 50%Sequential
Outline	Time_spent	Total time spent on Outline content (Based on total session duration)	>= 50%Global	< 50%Sequential

$$W_i = \begin{cases} \frac{t_i}{T_i} & \text{(For Video, Audio, Simulation)} \\ \\ \frac{t_i}{T} & \text{(For PDF, PPT, Doc, Summary, Outline)} \end{cases} \tag{6.1}$$

Where t_i= time spent on LO_i, T_i= total duration of LO_i, and T= total session duration. When more than 50% of a video is watched by a student, it is quite probable that they prefer active learning rather than reflective. Consequently, when a student spends more than 50% of their time on textual documents (e.g. pdf or doc documents), it can be presumed that they are more of a reflective learner than an active one.

Example of time spent calculation: Table 6.2 shows the learning styles of Victoria, Harry and Michelle calculated on the basis of the time spent on learning objects for Information input dimension (Visual/Verbal).

Table 6.2: An Example of Time spent

Student (time spent)	**LO (length)**	**W**	**Learning style**
Victoria (30 min)	video (40 min)	$30/40 = 75\%$	LS = Visual
Harry (30 min)	PDF (35 min)	$30/35 = 85\%$	LS = Verbal
Michelle (45 min)	Audio (60 min)	$45/60 = 75\%$	LS = Verbal

6.2.1.2 Number of messages

By means of the number of messages in the course discussion forum, a student's tendency for social orientation is indicated, if the student is an active or reflective learner, as presented in the following Eq. (6.2).

$$M = \frac{\textit{Number of messages sent by the student}}{\textit{Average number of messages sent during the sessions}} \tag{6.2}$$

In the case where no messages are exchanged during a specific session, the value of M is undefined. On the other hand, a M value greater than or equal to 1 implies that the student is an active learner, whereas a negative M value suggests a reflective learner. A greater M value positively suggests active learning of a much stronger level, due to the higher possibility of active students posting messages more frequently than passive ones.

Example of number of messages calculation: Table 6.3 illustrates the case where 3 students Victoria, Harry, and Michelle have exchanged 100 messages throughout the session, i.e. $100/3 = 33.3$ messages in average have been sent per student. Victoria is reflective because she has sent less than the average number of messages, while Harry and Michelle are active for sending more than the average number of messages during the session.

Table 6.3: An Example of forum discussion

Student (no. of messages)	**M**	**Learning style**
Victoria (25 messages)	$25/33.3 = 0.75$	LS =Reflective
Harry (40 messages)	$40/33.3 = 1.20$	LS =Active
Michelle (35 Massages)	$35/33.3 = 1.05$	LS =Active

6.2.1.3 Mapping LOs to FSLSM dimensions

The formats of the learning objects accessed by a student define the learning style of the student as in Table 6.4. The symbol "-" in yellow cells means that learning objects of that format are irrelevant to the corresponding leaning style attribute. The value 1 indicates the learning style attribute associated to that format in the learning style dimension. The value 0.5 indicates that both learning style attributes of the dimension are associated to that format. Both learning style calculation and student profile adaptation are described in the following sections.

Table 6.4: Mapping of Learning Objects format and activities as FSLSM

		Learning Style Dimensions							
		Information Processing		Information Perception		Information Input		Information Understating	
Learning object	LO\LS	Active	Reflective	Sensing	Intuitive	Visual	Verbal	Sequential	Global
Format	Video	1	0	-	-	0.5	0.5	-	-
	Audio	0	1	-	-	0	1	-	-
	Presentation (PPT)	0.5	0.5	0	1	-	-	0	1
	PDF	0	1	0	1	0	1	1	0
	Doc	0	1	0	1	0	1	1	0
Activities	Exercise	1	0	0.5	0.5	-	-	0.5	0.5
	GroupAssignment	1	0	-	-	-	-	-	-
	Individual Assignment	0	1	-	-	-	-	-	-
	Summary	-	-	-	-	-	-	0	1
	outline	-	-	-	-	-	-	0	1
	simulation	1	0	-	-	-	-	-	-
	forum	1	0	-	-	-	-	-	-
"1"	Relevant Positive Learning Object			"0"	Relevant Negative Learning Object				
"-"	Irrelevant Learning object			"0.5"	Relevant positive to two LS criteria within the same dimension				

6.2.2 Dynamic Profile Modeling

As mentioned in Chapter 2, a major limitation in relation to a number of current adaptive e-learning systems constitutes the issue where modelled student profiles are static or low-dynamic and that the underlying representation of the students learning styles are not evolving.

In this phase, we aim at introducing a novel algorithm for learning style adaptation, as shown in Figure 6.2. The proposed algorithm infers student behavioural patterns over K sessions in addition to computing the vector $KSSP$ which is the number of hits for all of the eight learning style attributes based on student learning behaviour patterns, as discussed in Section 6.2.1.

For instance, $KSSP[1]$ depicts the number of hits for the "active" learning style, while $KSSP[2]$ represents the number of hits for the "reflective" learning style, and $KSSP[8]$ is the number of hits

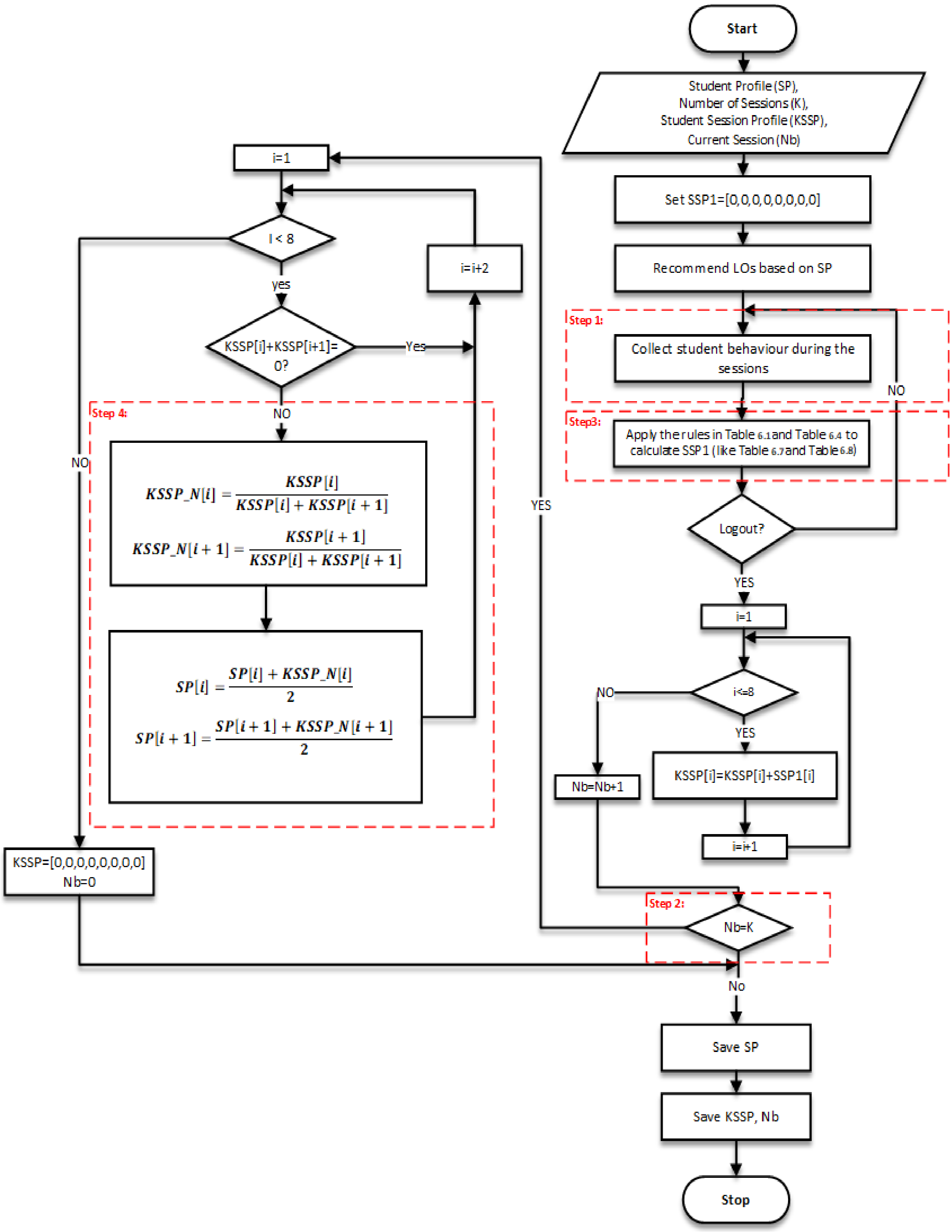


Figure 6.2: Updating student profiles dynamically flowchart

for the "intuitive" learning style, by making use of the same indexing as in (3.1) (see Section 3.1.1).

By the end of K sessions, a normalisation of the number of hits within each dimension as in Eq. (6.3) takes place so as to compute the current learning style vector *KSSP_N*.

$$\left.\begin{array}{l}
\textbf{for } i = 1:2:6:8 \textbf{ do} \\
\quad \textbf{if } (KSSP[i] + KSSP[i+1] \neq 0) \\
\quad\quad KSSP_N[i] = \frac{KSSP[i]}{KSSP[i]+KSSP[i+1]} \\
\quad\quad KSSP_N[i+1] = \frac{KSSP[i+1]}{KSSP[i]+KSSP[i+1]} \\
\quad \textbf{else} \\
\quad\quad KSSP_N[i] = 0 \\
\quad\quad KSSP_N[i+1] = 0 \\
\quad \textbf{endif} \\
\textbf{enddo}
\end{array}\right\} \tag{6.3}$$

Tables 6.7 and 6.8 demonstrate the way in which the vectors *KSSP* and *KSSP_N* are determined.

6.2.3 Profile updating

The final step is to update dynamically the student profile *SP* using the learning style *KSSP_N* calculated during the *K* sessions. This is done by calculating the new value of *SP* as in Eq. (6.4).

$$\left.\begin{array}{l}
\textbf{for } i = 1:8 \textbf{ do} \\
\quad \textbf{if } (KSSP_N[i] \neq 0) \\
\quad\quad SP[i] = \frac{SP[i]+KSSP_N[i]}{2} \\
\quad \textbf{endif} \\
\textbf{enddo}
\end{array}\right\} \tag{6.4}$$

That is for each learning style attribute i, $i = 1, 2, \cdots, 8$, if $KSSP\_N[i] \neq 0$, then a new learning style value is computed by considering the average between the previous and current learning style values; if not, the previous learning style value is maintained. The new student profile *SP* is utilised for recommending personalised learning objects to students the next time they log in, as explained in Chapter 5.

The following section covers an explanation of the working of the algorithm 6.2 by making use of a real academic case study.

6.3 Example

In this section, the modelling of a real academic example has been focused on demonstrating how the proposed algorithm works. As explained in Chapter 4, students from AAST, Egypt were required to fill-out the dynamic FSLSM questionnaire for the purpose of initialising their profiles. Table 6.5 shows instances of initial learning style vectors.

Table 6.5: Examples of student learning style vectors

	Active	**Reflective**	**Visual**	**Verbal**	**Sequential**	**Global**	**Sensing**	**Intuitive**
Victoria	0.4	0.6	0.25	0.75	0.5	0.5	0.8	0.2
Michelle	0.5	0.5	0.6	0.4	0.8	0.2	0.7	0.3

6.3.1 Profile adaptation scenario

The working of the student profile (SP) learning style adaptation algorithm is described below:

Step 1*– Recommendation of course LOs based on their similarity to SP learning styles using hybrid recommendation algorithm, as explained in Chapter 5.* Table 6.6 shows recommended LOs to Victoria and Michelle through 5 sessions.

Step 2 *–Collect student behaviour during K sessions (in this case K=5).* A sample of learning behaviour patterns of Victoria and Michelle is presented in Table 6.6.

Step 3 *-Applying adaptation rules as illustrated in Table 6.1 and Table 6.4.* The algorithm first computes the total time spent by Victoria and Michelle on Learning objects (refer to Section 6.2.1) followed by computing the number of messages sent by them throughout the session, as covered in Table 6.6. The average number of messages is $200/8 = 25$, where 8

Table 6.6: Examples of Students' behaviour patterns

	Learning object		Victoria's Learning behaviour patterns		Michelle's Learning behaviour patterns	
	Format	LO total time	Victoria's spent time	Session Duration	Michelle's spent time	Session Duration
Session 1	ppt		30 mins	75 mins	45 mins	60 mins
Session 2	Outline		10 mins	30 mins	0	0
Session 3	Video	40 mins	30 mins		20 mins	
Session 4	Summary		0	0	15 mins	32 mins
Session 5	PDF		45 mins	80 mins	10 mins	50 mins
		Total Number of messages during session	Messages sent by Victoria		Messages sent by Michelle	
Discussion	Forum Discussion	200	30		18	

Table 6.7: KSSP Calculation For Victoria's behaviour

Victoria's behaviour		**Active**	**Reflective**	**Visual**	**Verbal**	**Sequential**	**Global**	**Sensing**	**Intuitive**
PPT=$30/75 = 0.4$	PPT $< 50\%$:	0.5	0.5			1		1	
Outline=$10/30 = 0.35$	Outline$< 50\%$				1	1			
Video=$30/40 = 0.75$	Video$>= 50\%$	1		0.5	0.5				
Summary= 0	No behaviour								
PDF=$45/80 = 0.56$	PDF $>= 50\%$		1		1	1			1
Messages $= 30/25 = 1.2$	M $>= 50\%$	1							
KSSP Calculation									
KSSP		2.5	1.5	0.5	2.5	3	0	1	1
KSSP_N		0.63	0.37	0.17	0.83	1	0	0.5	0.5

is the total number of participants during the forum discussion (refer to Section 6.2.1). The values of KSSP and KSSPN are then determined, as demonstrated in Table 6.7 and Table 6.8, for Victoria and Michelle, respectively.

Table 6.8: KSSP Calculation For Michelle's behaviour

Michelle's behaviour		**Active**	**Reflective**	**Visual**	**Verbal**	**Sequential**	**Global**	**Sensing**	**Intuitive**
PPT=$45/60 = 0.4$	PPT $< 50\%$	0.5	0.5			1		1	
Outline= 0	No behaviour								
Video=$15/40 = 0.37$	Video$< 50\%$		1	0.5	0.5				
Summary= $15/32 = 0.46$	Summary$< 50\%$					1			
PDF=$10/50 = 0.20$	PDF $< 50\%$	1		1			1	1	
Messages =$18/25 = 0.72$	M $< 50\%$		1						
KSSP Calculation									
KSSP		1.5	2.5	1.5	0.5	2	1	2	0
KSSP_N		0.37	0.63	0.75	0.25	0.67	0.33	1	0

Step 4 – Update Victoria and Michelle 's profiles with the new learning styles as presented in Table 6.9 and Table 6.10.

A significant question remains, however: "*is the proposed profile adaptation algorithm effective in*

Table 6.9: Victoria 's Updated learning style profile

	Active	Reflective	Visual	Verbal	Sequential	Global	Sensing	Intuitive
Victoria's SP	0.4	0.6	0.25	0.75	0.5	0.5	0.8	0.2
KSSP_N	0.63	0.37	0.17	0.83	1	0	0.5	0.5
Victoria's new SP	0.52	0.48	0.21	0.79	0.75	0.25	0.65	0.35

Table 6.10: Michelle's Updated learning style profile

	Active	Reflective	Visual	Verbal	Sequential	Global	Sensing	Intuitive
Michelle's SP	0.5	0.5	0.6	0.4	0.8	0.2	0.7	0.3
KSSP_N	0.37	0.63	0.75	0.25	0.67	0.33	1	0
Michelle's new SP	0.44	0.56	0.68	0.32	0.73	0.27	0.85	0.15

predicting student learning styles?" An experimental study which was conducted aimed at finding an answer to this question.

6.4 Experiment: Examining the effectiveness of the proposed algorithm in predicting student learning styles

In this section, the main goal is to validate our proposed dynamic student profile adaptation. An experimental study was conducted for examining the effectiveness of our proposed algorithm in relation to the updated SP accuracy. Hence, the aim of this experiment was to show that our approach can dynamically capture students' LSs using their behaviour patterns as well as be used in e-learning recommendation systems for the development of accurate and effective learning objects. In this experiment, we will apply the predicted rating algorithm explained in Section 5.1.1 in Chapter 5 to predict students' rating.

On the basis of the questionnaire responses, or initial student profiles (SL_{1s}), we then compare students' actual ratings of learning objects with the predicted ones on the basis the adapted student profile (LS_{2s}). For this purpose, two metrics - the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) have been made use of for evaluating and measuring the preciseness of the adaptation algorithm in addition to the learning styles detection as explained in Chapter 5. The profile adaptation algorithm was implemented in C++ through means of the Visual Studio and Windows Presentation Foundation (WPF) for designing a Graphical User Interface (GUI), as illustrated in Fig. 6.3. Moreover, all experiments were run on a Windows-based PC with an Intel core i5 processor having a speed of 2.40 GHz and 16 GB of RAM. The following section presents the results of the experiments.

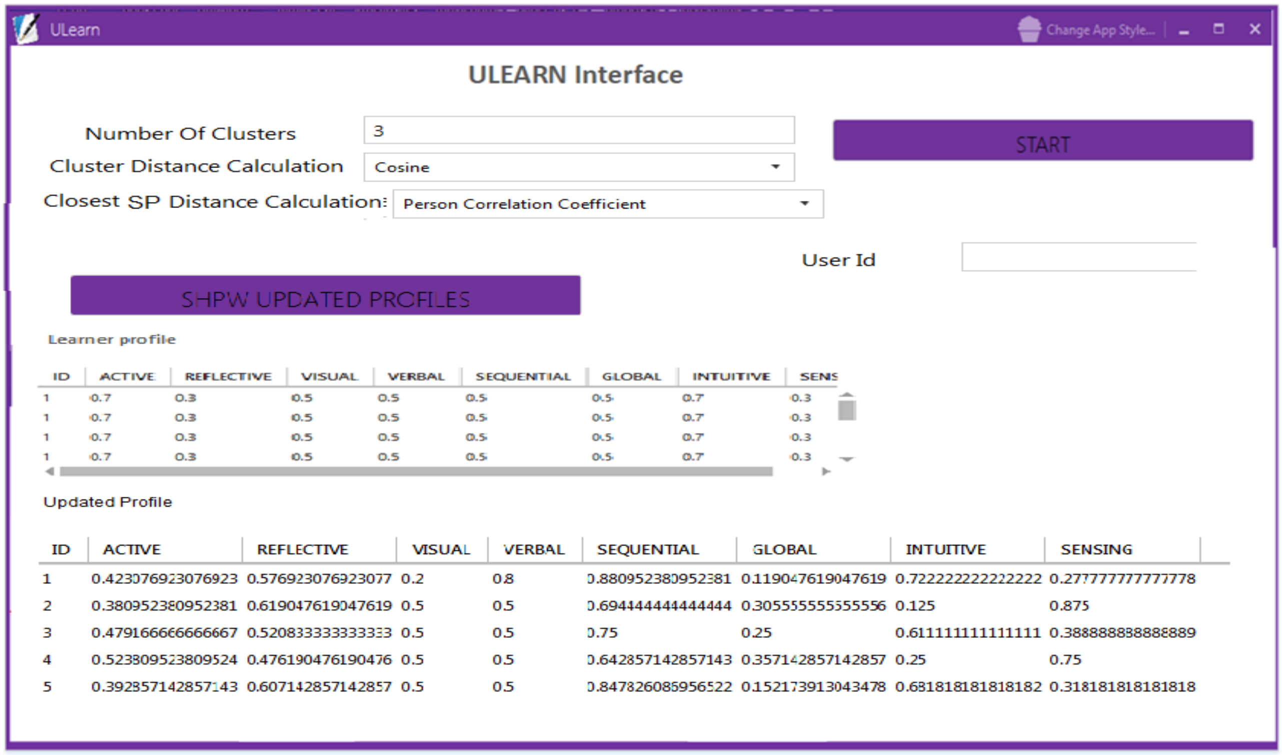


Figure 6.3: Dynamic student profile adaptation interface

6.4.1 Experimental results and discussion

Following the experiment, it was inferred that the proposed algorithm is capable of significantly improving prediction results within all learning styles dimensions for a random sample of 80 students whilst studying three topics, each with three lessons, which wasn't the case with the ILS. On the basis of these observations, a conclusion that the monitoring of learners' behaviours with time spent on different learning object formats can enhance the accuracy of detecting students' learning styles within an e-learning recommendation system was derived.

All in all, this adaptation algorithm can assist the system in improving student performance by recommending appropriate course learning objects to match their learning styles. Furthermore, experimental results demonstrated that the student rating prediction algorithm possesses the best accuracy in cases where the SP adaptation takes places through behaviour ($LS2s$) instead of the

ILS questionnaire ($SL1s$), as illustrated in Fig. 6.4 and Fig. 6.5 where MAE and RMSE are respectively used as measurement tools.

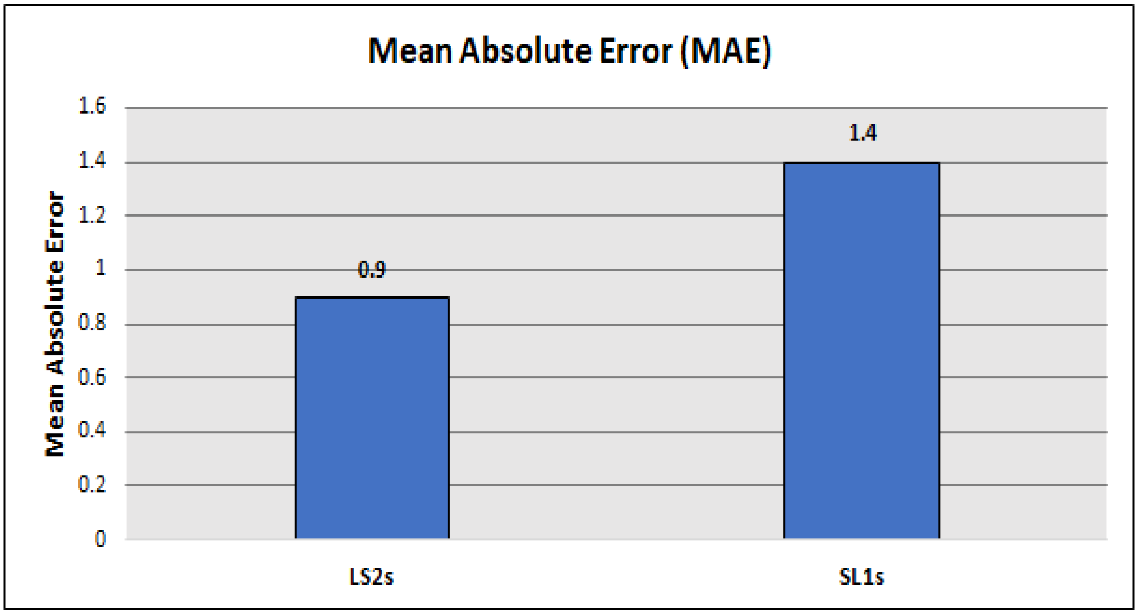


Figure 6.4: Adaptation algorithm accuracy using MAE

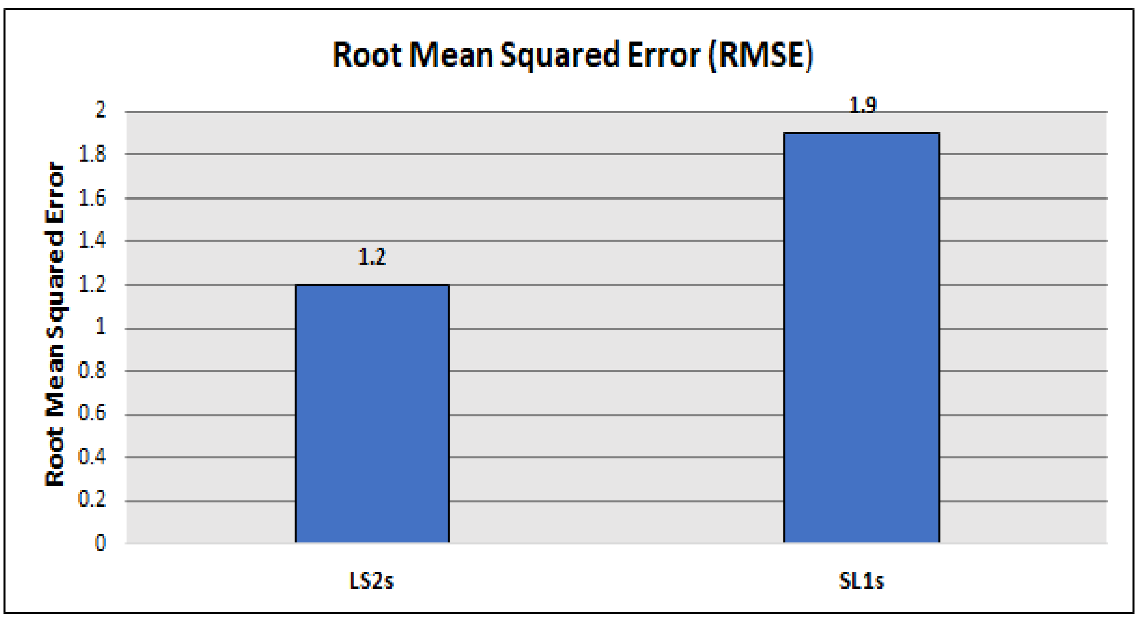


Figure 6.5: Adaptation algorithm accuracy using RMSE

6.5 Summary

In this chapter, we presented new algorithms for constructing dynamic student profile according to learning behaviour patterns. Unlike other works in the literature that built static student profiles, this chapter comprises the introduction of an algorithm for computation of a dynamic student profile in line with the FSLSM. The aim of this algorithm includes building and regularly updating student profiles on the basis of student behaviours throughout an online course and comprises of three crucial steps: 1) extraction of student learning behaviour patterns which reflect learning styles from the behaviour log file; 2) capturing student learning styles through quantitative methods; and 3) dynamic updating of student learning styles after each topic. On this account, the results from the experiment suggest the enhanced accuracy of the proposed algorithm compared to that of an ILS questionnaire. In the next chapter, the prototype system designed on the basis of the purposed e-learning recommendation ULEARN framework is presented. Furthermore, all the modules of the system prototype are discussed.

Chapter 7

Implementation and Evaluation of ULEARN recommender system

As described in Chapter 3, the architecture of ULEARN comprises three main layers (refer to Fig. 3.2). The second layer which is the Adaptive-recommendation layer contains three components, namely the Learning Styles initialisation Module (LS Module), the Learning Objects recommendation Module (LO Module) and the Dynamic Student Profile adaption Module (DSP Module), which have been implemented and evaluated respectively in Chapters 4, 5, and 6. Furthermore, the design and implementation of the other two layers (Presentation layer and Data layer) of the ULEARN architecture will be covered in this chapter in addition to an evaluation of the overall system with respect to student satisfaction and system usability.

The remainder of this chapter is structured as described. First, Section 7.1 covers the ULEARN prototype in detail along with its modules. On the other hand, Section 7.2 focus on system testing, whereas Section 7.3 covers the experimental design of evaluating the ULEARN system. Finally, Section 7.4 wraps up the chapter in terms of a summary.

7.1 ULEARN recommender system implementation

In this section, we introduce a prototype ULEARN recommendation system which we have implemented as a client-server application. Implemented in C++ using .NET environment and Windows

Presentation Foundation (WPF), the ULEARN system is a dynamic e-learner recommender system. The WPF was chosen over Windows Forms owing to the former being a larger comfortable technology than Windows Forms (preceding technology). In terms of the user interface of the recommender system, eXtensible Application Markup Language (XAML) was made use of. The prototype application was implemented and run on Windows-based PC with an Intel(R) core i5 processor having a speed of 2.40 GHz and 16 GB of RAM, under Windows 10. Moreover, Figure 7.1 illustrates the ULEARN e-learning sequence and activity diagrams to express the student's interaction with the ULEARN prototype, particularly in the course of the registration process, logging in into the ULEARN, filling-out the dynamic ILS questionnaire, learning object recommendation and updating learning styles. Tables are designed and created to store the entities described in Section 7.1.1. A detailed description of the prototype architecture is covered in the subsequent sub-sections.

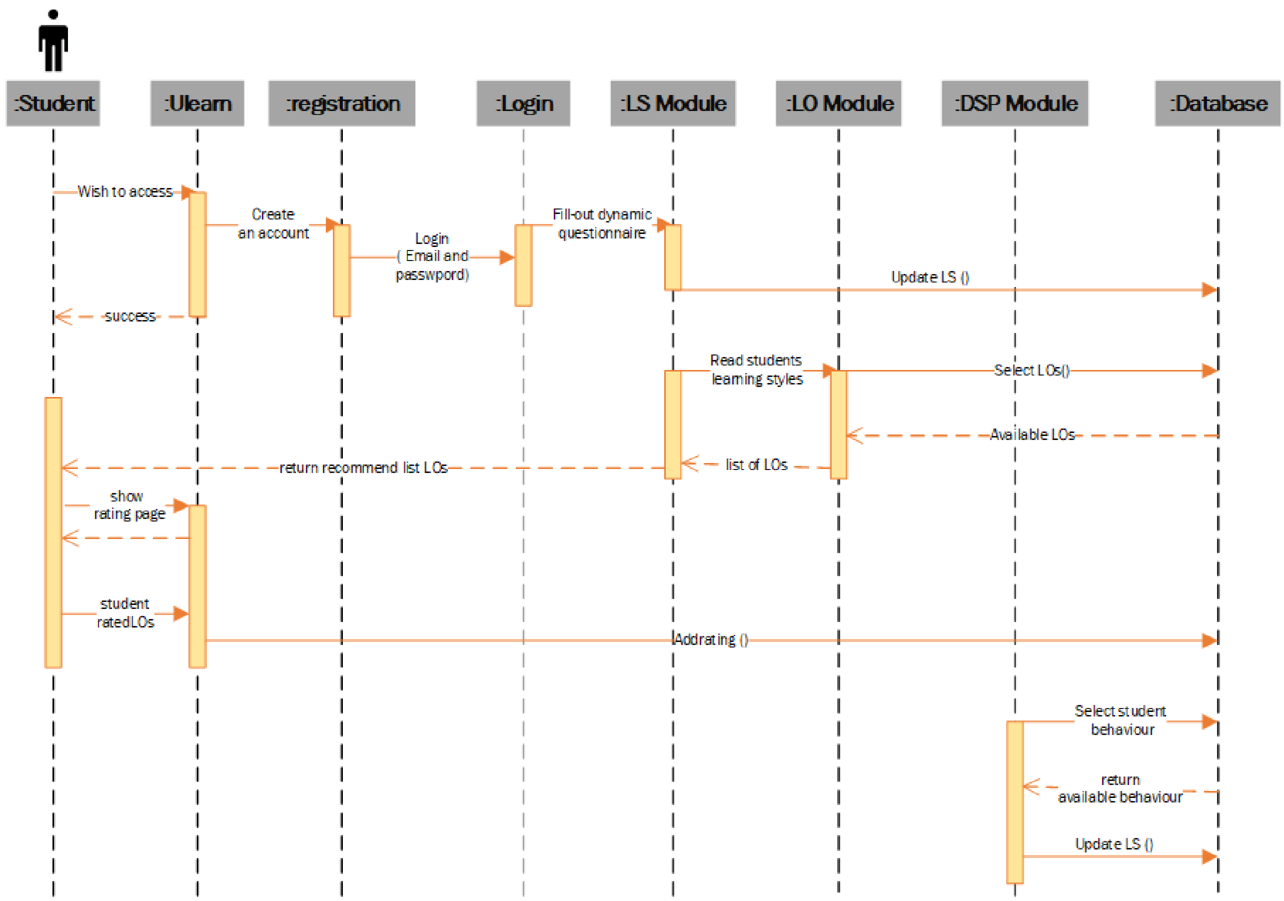


Figure 7.1: ULEARN prototype sequence diagram

7.1.1 Database Model

This section provides details of the database design. There are two ways to build a database, namely, Database-First and Code-First [312]. The Database-First, comprises modelling of a pre-existing database (for example, creating tables and relationships using the SQL server), while the Code-First mode involves the initial creation of the entity classes. Fig. 7.2 illustrates an example of student classes. Following this, the Entity Framework Core runtime will create a database with necessary tables from these entity classes. Once a database is created, it is synchronised with any changes in the code, so that no manual changes need to be added.

EXAMPLE OF STUDENT CLASS DEFINITION

```
public string Name {get;set;}
public string Email {get;set;}
public string Password {get;set;}
public string Phone {get;set;}
public string PhoneVerificationCode {get;set;}
public string EmailVerificationCode {get;set;}
public string PasswordVerificationCode {get;set;}
[ForeignKey("LearnerId")]
public Learners Learners {get;set;}
public ICollection<CurrentLessons> CurrentLessons {get;set;}
public ICollection<RefreshToken> RefreshToken {get;set;}
```

Figure 7.2: Example of Student Class Definition

As mentioned in section 3.2.2, The ULEARN system database stores all the data, which includes the following main components:

- **Student Profiles:** Students profile information saved in three main tables, as shown in class diagram Fig. 7.3. User basic Information table records the user ID and other attributes (e.g. Name and email address, etc.). Whereas, a numerical value for each learning style criteria (which has been calculated from the dynamic questionnaire or learning behaviour) is saved in the Learner table. The learning styles are held in eight values, one value for each learning style criteria, as described in Chapter 3. Furthermore, the learner History table records the previous learning styles profiles to be used while calculating the new profile, as explained in Chapter 6.

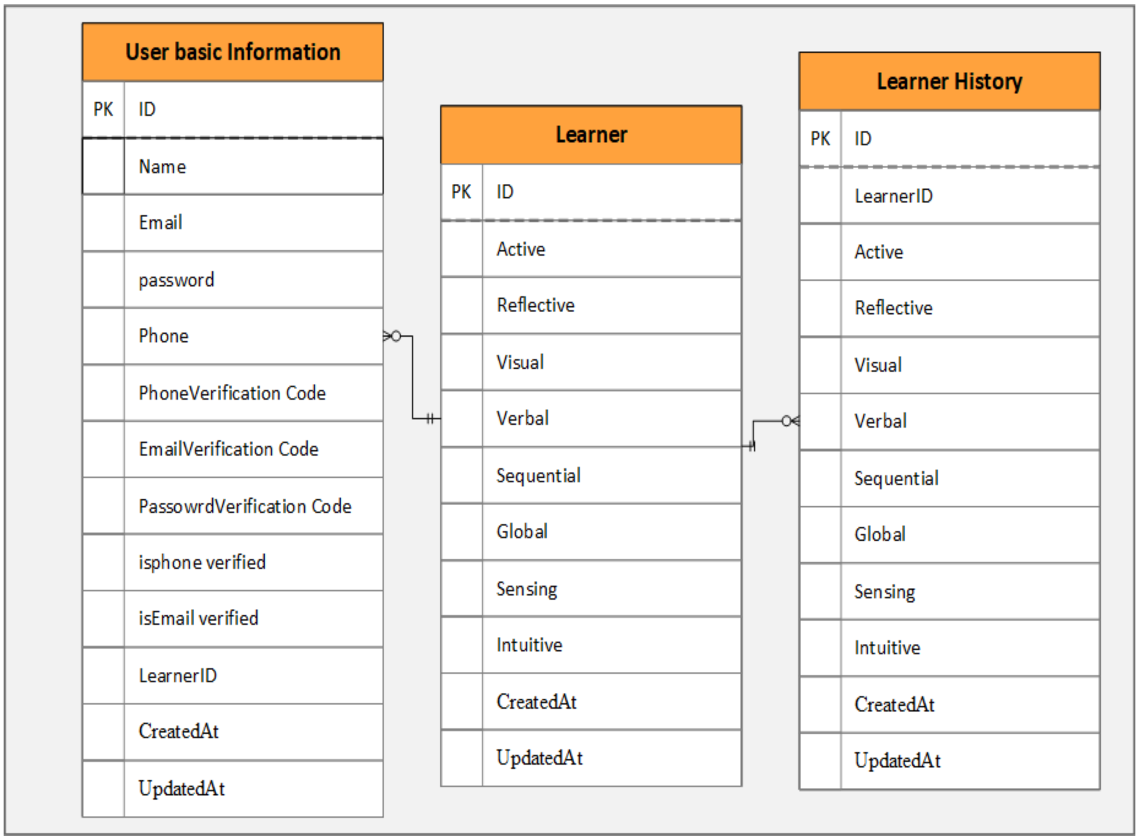


Figure 7.3: Student's Profile Class Diagram

– **Learning behaviour:** As explained in section 3.2.2, it is necessary that the ULEARN system also saves students' learning behaviour patterns during sessions so as to be able to update their profiles (learning styles). Fig. 7.4 shows the Behaviours table which records students' behaviour patterns during studying (e.g. NumberOfAccess, TotalLearnerSpentTime, and NumberOfMessages) in order to calculate their learning styles.

– **Learning object:** The *Learningobjects* table has a record object profile (learning styles), whereas *LearningobjectsInfo* table includes the learning objects path in the server and other attributes (e.g. TotalTime), as shown in Fig. 7.5.

– **Student rating:** Furthermore, all students' ratings will be saved in a separate table in the database named *Learner_LO_Rating* and linked to the learner by the learner ID as shown in

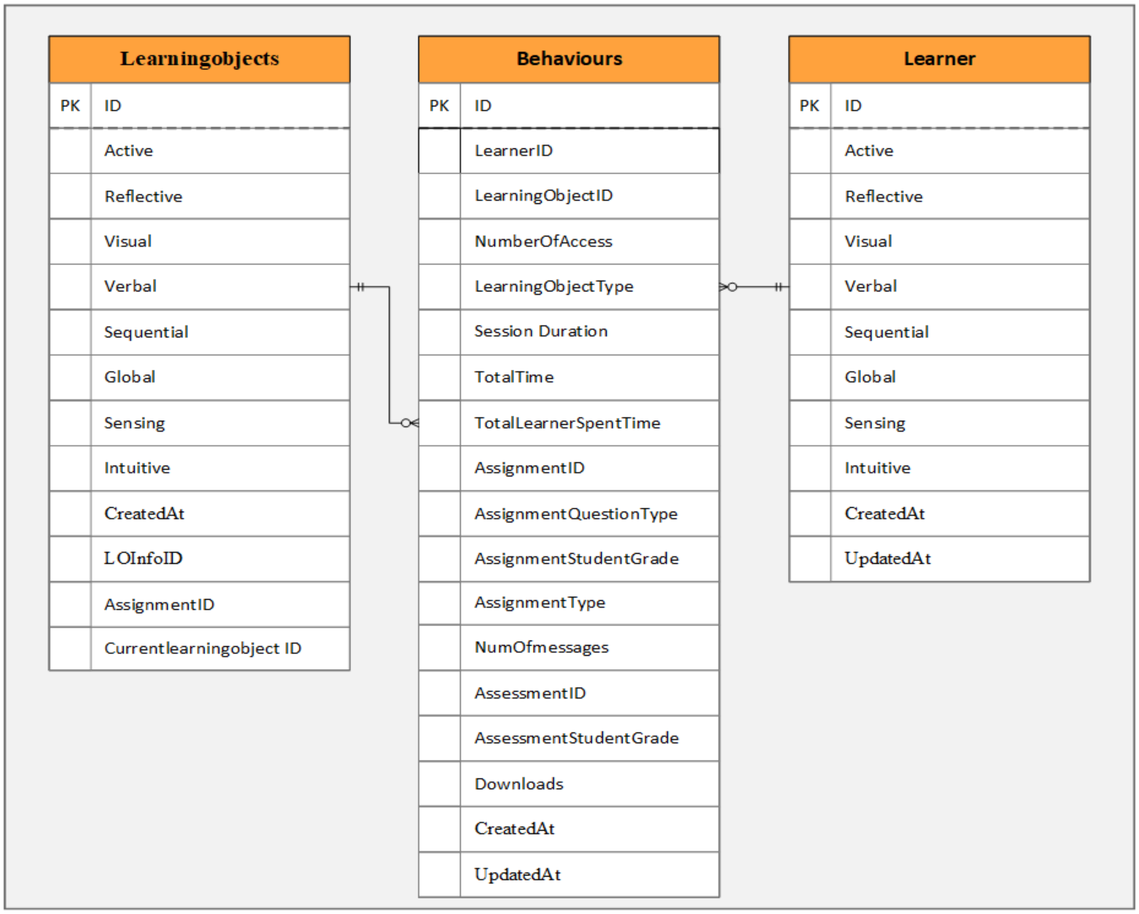


Figure 7.4: student's Behaviour Class Diagram

Fig. 7.6.

The next section will explain the system Interfaces and all its features in detail.

7.1.2 ULEARN Interface

As previously mentioned, the user interface within the ULEARN system are implemented using WPF using XAML language. This system was designed in such a way so as to be able to follow the general guidelines of usability [313] in addition to being easy-to-use, functional, friendly and personalised. The first thing the user comes across when accessing the application is the welcome page, as shown in Fig. 7.7. As explained in Chapter 3, ULEARN provides a user interface for students, teachers and administrators. First, Fig. 7.8 expresses the students' interaction use-case with the system, permitting students to register, log into the system, study Learning Objects and communicate with other students by sending messages through forums. Second, Fig. 7.9 expresses the teacher's interaction use-case at the beginning of using the system where teachers can log into

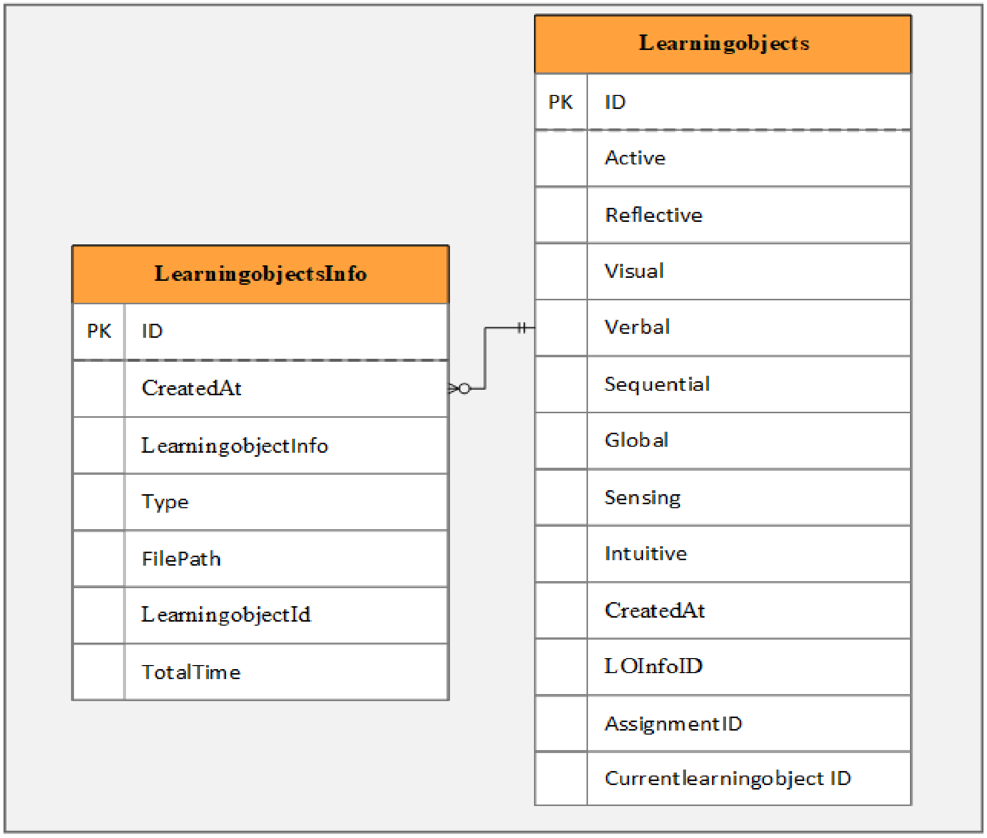


Figure 7.5: Learning object's profile Class Diagram

the system using this interface, create courses, upload learning objects, create objects profile for each LO, upload assignments, and send messages to students. The screenshot of the teacher's Graph User Interface is presented in Appendix D Fig D.3.

Illustrated in Fig. 7.10 is the Administrator's Use Case Diagram where the administrator can assign roles to users, manage teachers' information, assign courses to teachers, manage course information, manage lessons for the course in addition to management of the system database. As an initial step, students register themselves in the ULEARN portal before being able to use the system. During this registration process, personal details including the full name, email address, password, educational level (e.g. undergraduate) and major (e.g. computer science) are gathered, as shown in Fig. 7.11.

After completion of the registration process, a student authenticates his/her profile using username and password credentials. The MD5 algorithm is made use of to hash the original passwords instead of storing it as plain text in the database. Similarly, during authentication, the input password is hashed by MD5, followed by the comparison of the resultant hash value with the stored value in the database for that specific user. An example of the way in which hashed passwords appear

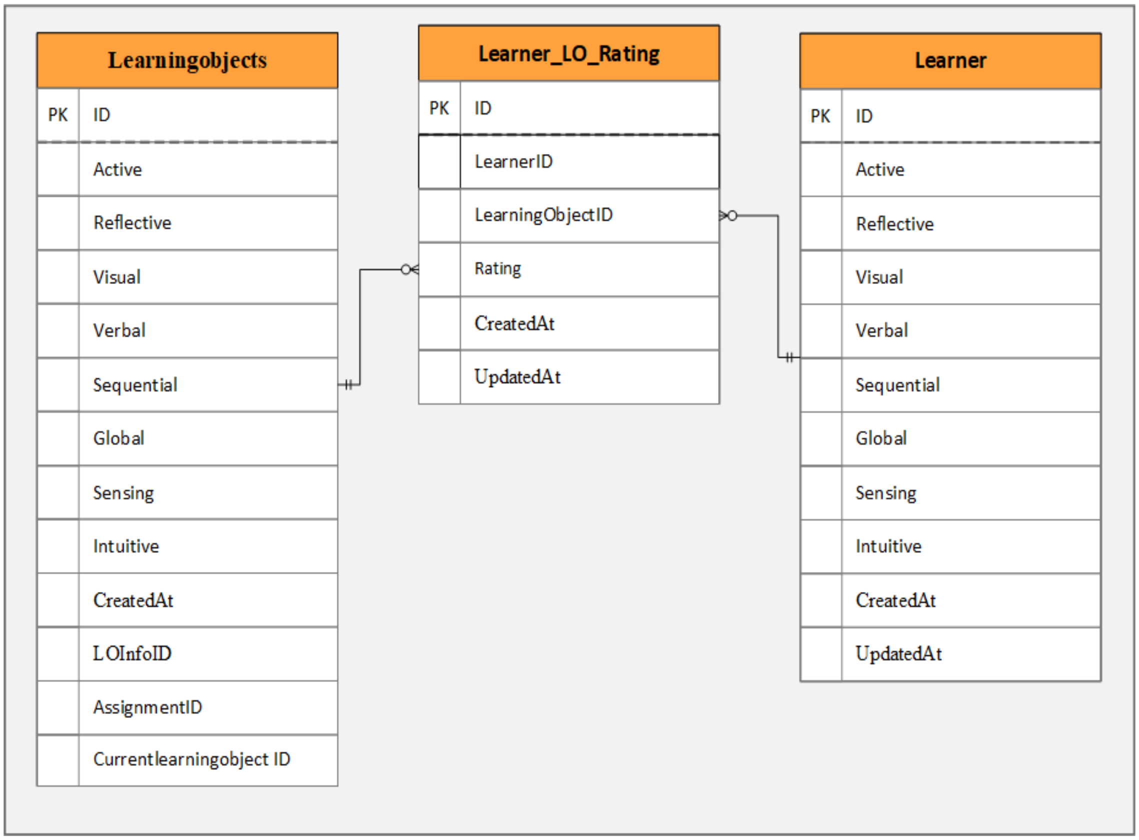


Figure 7.6: Students rate Class Diagram

in the database is shown in Fig. 7.12. After the successful authentication process, the student asked to fill-out the Dynamic ILS questionnaire to initialise their learning styles profile. As soon as student submit the ILS questionnaire, s/he will be directed to the recommendation page, which will contain learning objects recommendations based on their learning styles.

7.1.3 Student Learning Style initialisation

For the purpose of determining the initial learning style of a student, s/he is required to fill-out the Dynamic ILS questionnaire that is provided as part of the ULEARN system (see Chapter 4). After completing the questionnaire, the Learning styles initialisation module will initialise students' learning styles which are saved in the Learner table ULEARN Database. The questionnaire interface looks like in Fig. 7.13.

After Initialising a student profile, the system will recommend learning objects to students, as explained in the following Section 7.1.4.

Figure 7.7: ULEARN Welcome Page

7.1.4 Learning objects Recommendation

The main aim of the learning objects recommendation module comprises recommending students with suitable learning objects to match their learning styles in order to reduce information overloading. To implement the proposed approach in this thesis, a database stores learning objects with different formats (e.g. ppt, pdf, video and docs etc.). The ULEARN is flexible which means it could be applied in a different domain. In this study, undergraduate was chosen as the qualification level sample in the implementation which will be explained in Section 7.3.

When students log-in to the ULEARN system, they will be asked to select which course they want to study depending on the major selected during registration. Following this, the system recommends the most suitable learning objects for this lesson that matches the students' learning styles from the database. Figures 7.14 and 7.15 present a screenshot of a personalised LOs recommendation list. The recommendation interface includes three tabs, namely, recommendations, my lessons and history.

The recommendation interface includes three tabs, namely: recommendations, my lessons and history. The Recommendation tab includes recommend all lessons that have been recommended

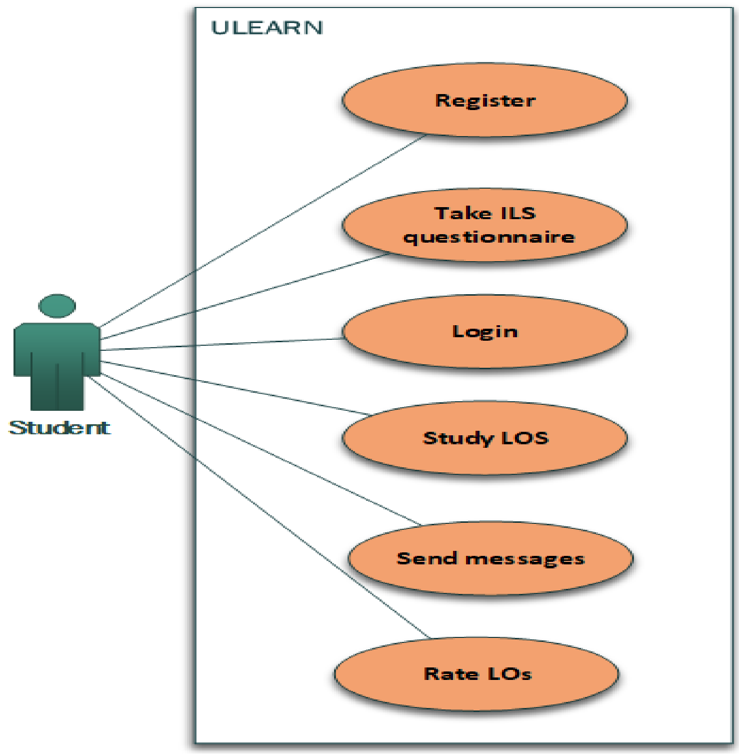


Figure 7.8: Use case diagram for student

according to the students' learning styles, as shown in Fig. 7.14.

While, my lessons tab includes a list of recommended learning object within the lesson that student currently studying, as shown in Fig. 7.15.

Finally, the history tab includes students progress such as all LOs that have been studied by students in the previous sessions, LO type, students grades and total time spent on LO, as shown in Fig. 7.16.

After students complete learning object the system will ask them to rat recommended LOs. A rating is achieved (on a Likert scale of 1-5) for each LOs, as presented in Fig. 7.17.

7.1.5 Student behaviour monitoring

In order to update students learning styles, the system collects a student's behaviour while they used the ULEARN system, all the activities they performed after login to the system, such as actions, time spent, number of massages and accessed objects, were recorded in table behaviours in ULEARN database (see chapter 6).

As we can see from Fig. 7.18 ULEARN system records session time while s/he studying LOs to

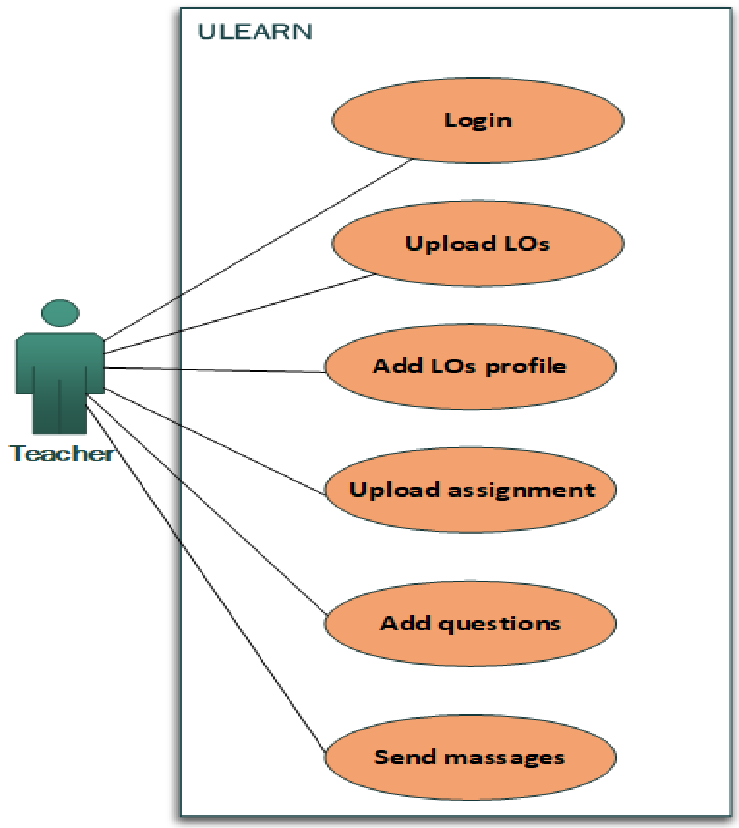


Figure 7.9: Use case diagram for teacher

be able to detect s/he learning styles and update their profile.

To calculate student learning styles from the number of messages, the proposed system is built an online chatting between students using SignalR library which is a technology developed by Microsoft technology in the year 2014 [314]. SignalR handles connection management automatically and let's broadcast messages to all connected clients simultaneously. This is a communication tool privately used by a group project. As was earlier mentioned, there are some students are struggling with communication skills, they prefer to work individually or in small groups, as shown in Fig. 7.19.

Therefore, ULEARN tries to improve communication among the student's group members. Then during students' commendation together, the system monitors students interactions through the number of messages and updates their learning styles.

Finally, the system will collect students behaviours to update their learning styles.

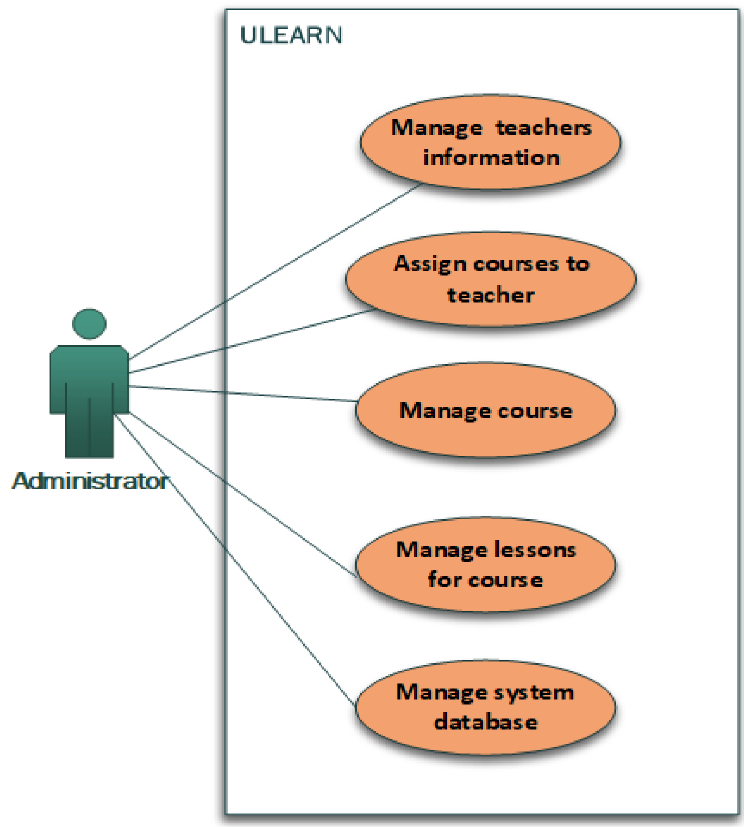


Figure 7.10: Use case diagram for Administrator

7.1.6 Student profile adaptation

After student's behaviour patterns have been monitored during different sessions. The dynamic student profile adaptation module starts to retrieve students behaviours and studied learning objects from the database. Then it calculates the total time spent and the number of messages to determine the new learning styles of student and update his/her LSs automatically considering previous student's learning style, which is then stored in history table in the database.

In the following sections, system testing is discussed to verify that the entire ULEARN system works according to students' preference.

7.2 System Testing

For testing ULEARN system, three testing methods are used with the goal of determining whether or not it meets the defined requirements like it's designed to (see Section 1.4). The testing methods are *Module testing*, *System testing* and *Usability testing*. The following sub-sections will describe

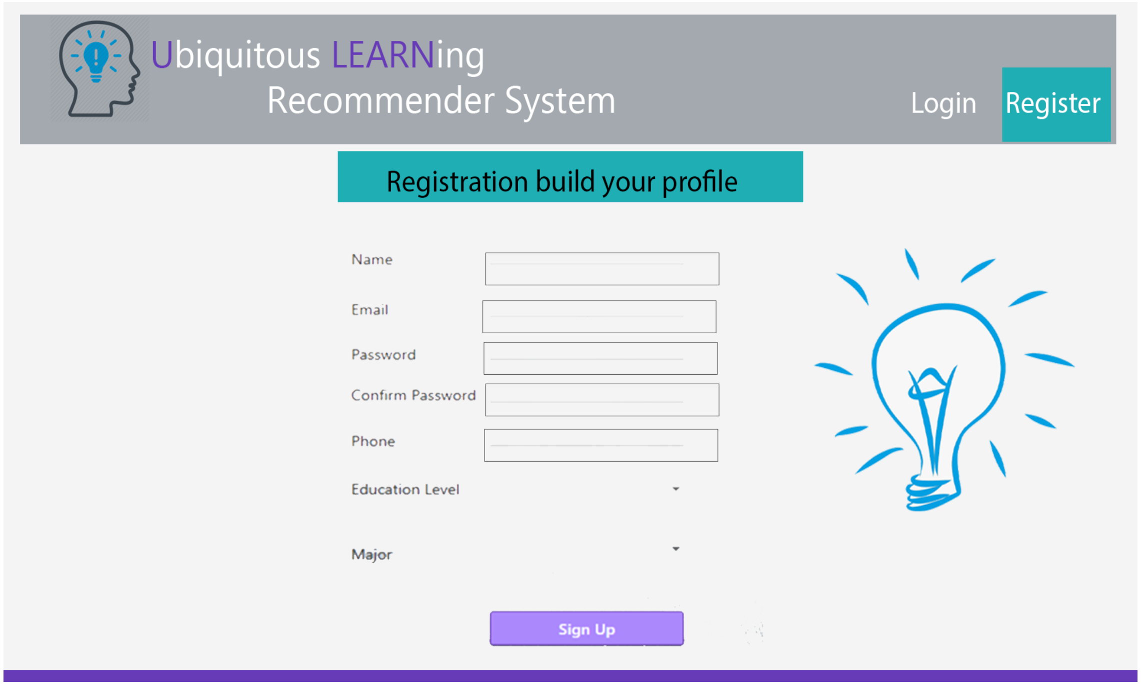


Figure 7.11: Create an account in ULEARN system

DESKTOP-E97MRLJ....arnDB - dbo.Users

Id	CreatedAt	UpdatedAt	Name	Email	Password
1	2019-08-13 10:5...	2019-08-13 10:5...	shima	shima@gmail....	%�Ž�@�d�mq<��
2	2019-08-24 07:5...	2019-11-15 01:4...	radwam	radwan@gmail....	%�Ž�@�d�mq<��
3	2019-08-24 08:0...	2019-08-24 08:0...	radwam	amal@gmail.C...	%�Ž�@�d�mq<��
4	2019-08-24 08:1...	2019-08-24 08:1...	radwam	moza@qmail.C...	%�Ž�@�d�mq<��

Figure 7.12: Example of hashed passwords in a database using MD5

these methods in detail.

7.2.1 Module testing

For *Module testing*, all modules (algorithms, classes, and functions) were tested carefully by checking every statement written. Module testing is used to test each feature of the system provided the input and expected output of the application. For the testing of the proposed modules three phases are achieved, as shown in Table 7.1.

7.2.2 system testing

After performing module testing, the *system testing* is conducted to test a fully integrated system to ensure whether it meets the requirements. In this research, we are using Functionality Testing

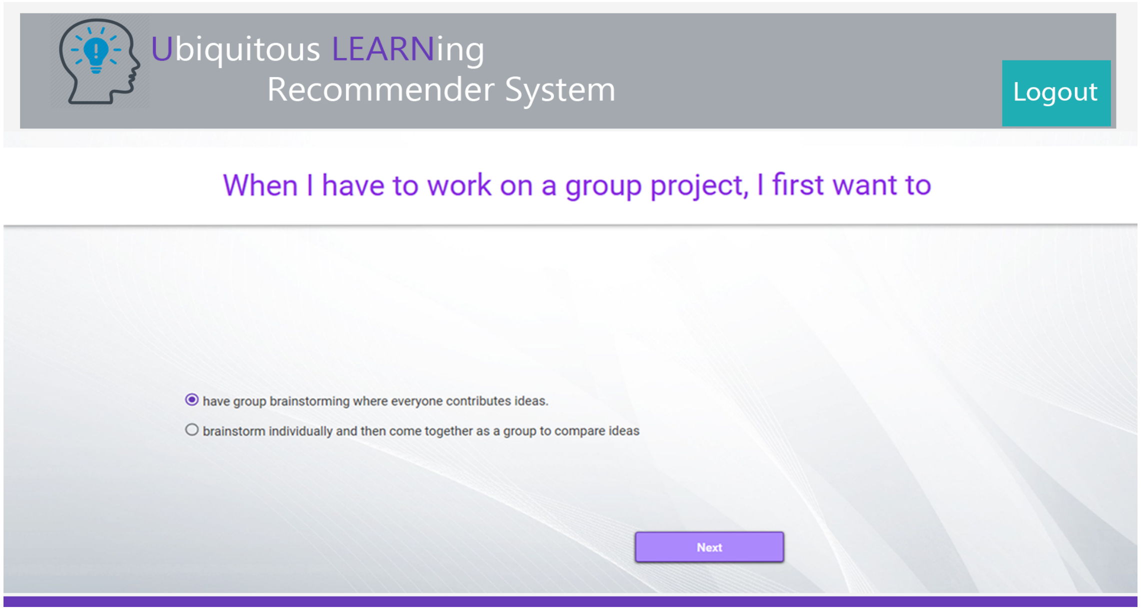


Figure 7.13: Dynamic ILS questionnaire

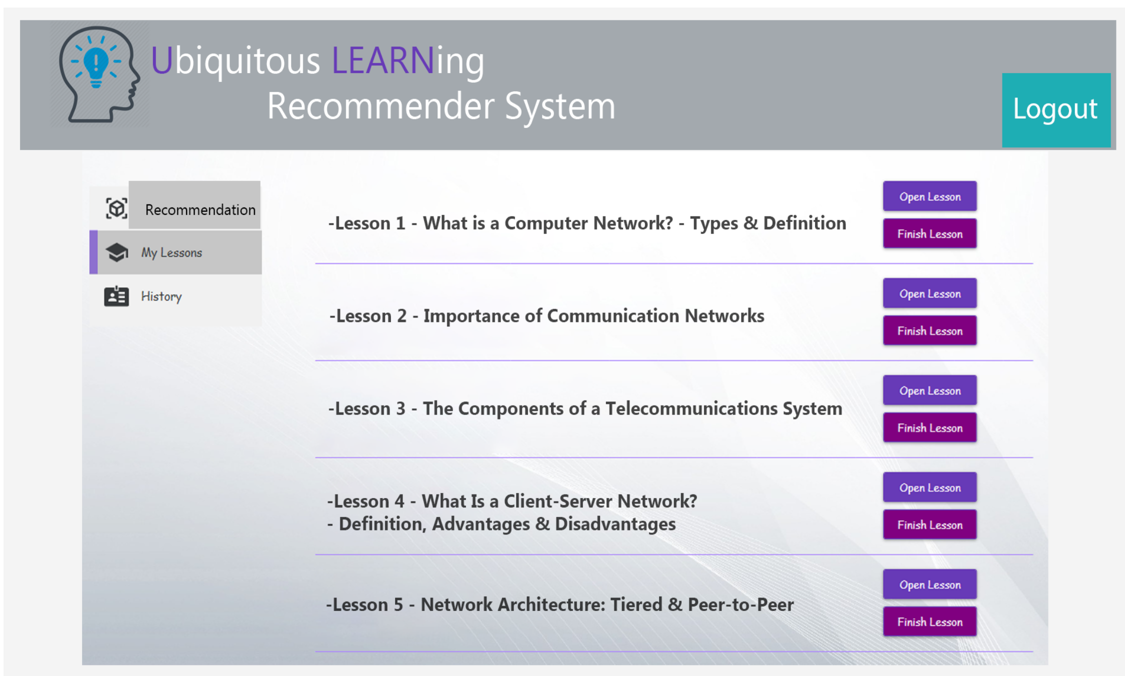


Figure 7.14: Learning object Recommendation

(FT). FT is a form of black-box testing that helps us to evaluate and assess the ULEARN system's performance [315]. The main importance of black-box testing is to handle both valid and invalid inputs from user perspective [316]. Table 7.2 shows the functional testing (test cases) used for testing student's interface functionalities. The details of test cases used for testing teacher's and administrator's portal are presented in appendix E.

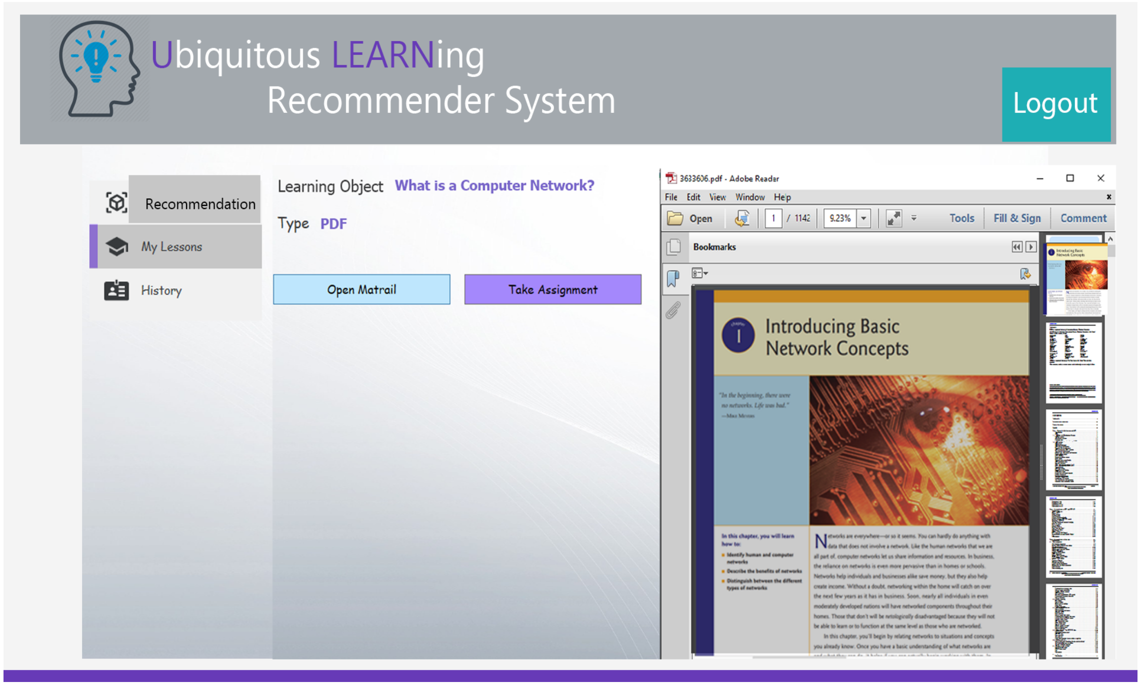


Figure 7.15: My lesson (Current learning object)

Table 7.1: Module Testing

Module Testing					
Testing Sequence	**Module Testing**	**Inputs**	**Tests**	**Expected Results**	**Status (Pass /Fail)**
1	LS Module (chapter 4)	A real academic case study (Students filled out LS dynamic questionnaire)	Test to see if LS Module can initialise learning styles.	The expected result is that initial learning styles saved in ULEARN database. (see Section 4.5).	Pass
2	LO Module (chapter 5)	Students' datasets from the AAST-MOODLE log-file.	Test to see if LO Module can recommend LO correctly.	The expected result is a list of the most suitable learning objects on the basis of student learning styles. Testing revealed that EHF algorithm achieved the highest accuracy in comparison to ECF and ECBF (see Section 5.2.4.1).	Pass
3	DSP Module (chapter 6)	Students' datasets from the AAST-MOODLE log-file.	Test to see if DSP Module can identify and update students learning styles automatically.	The expected result is that learning styles identified and Updated learning styles saved in ULEARN database (see Section 6.4).	Pass

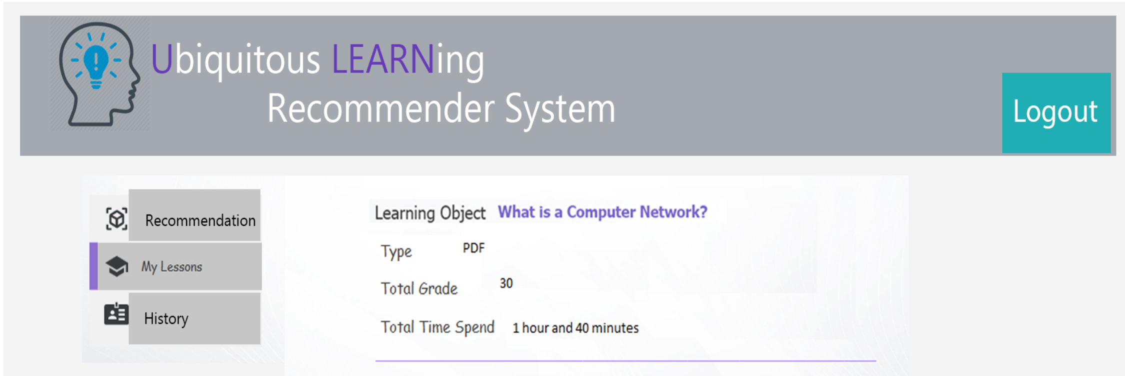


Figure 7.16: History (student progress)

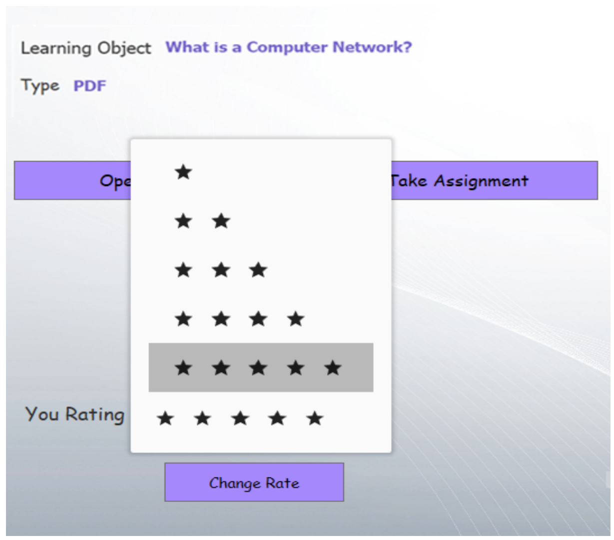


Figure 7.17: Student's rating

Through system testing, it found that the system is consistent with the expectation, which effectively improves the efficiency of identifying and updating student learning styles. Besides, improving the accuracy and reliability of recommendation of the learning object.

Eventually, student satisfaction and system usability testing for the proposed system will be conducted to measure program performance in user-centred design (Section 7.3). Student satisfaction and usability testing are considered a research tool, a fundamental evaluation method which enables to assess the proposed system performance [317]. The next section addresses the evaluation of ULEARN system in details.

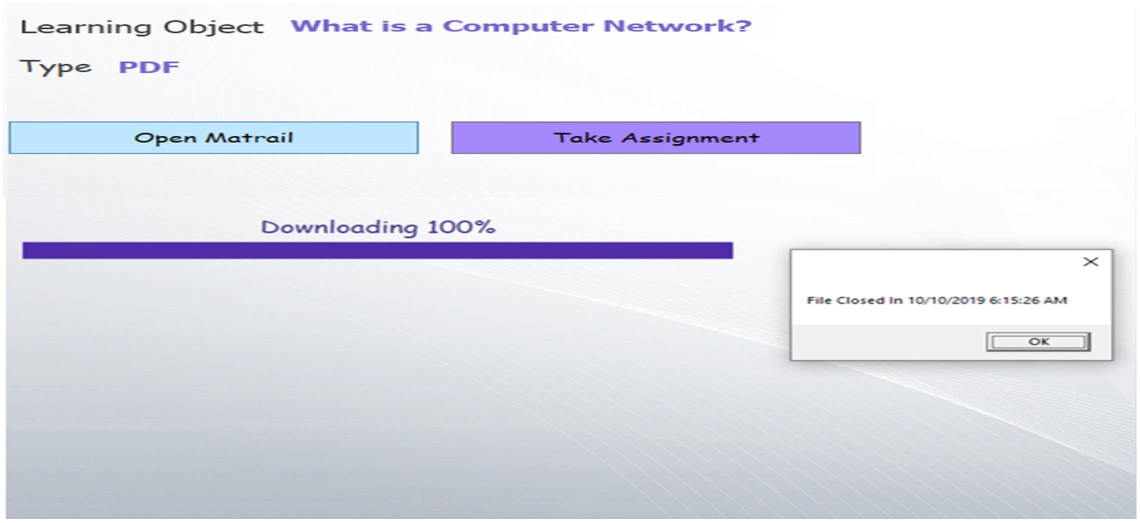


Figure 7.18: Time spent

7.3 ULEARN system evaluation

The proposed ULEARN recommender system's performance is evaluated and tested in this section through means of an experiment conducted in the School of Business at the Arab Academy for Science and Technology (AAST), Egypt. For this purpose, 70 undergraduate students were invited to participate in the evaluation, in particular, students who registered for the "Network and communication" course. This course comprised three lessons, with each lesson including five learning objects in various presentation styles (e.g. video, audio, PowerPoint, Word documents, and Pdf). Table 7.3 shows the sequence of lessons within sessions.

7.3.1 Student satisfaction

This experimental evaluation was conducted to measure student's satisfaction with respect to recommended learning objects and personalisation. Participants of this study were required to give their opinion about the system by filling out an E-Learner Satisfaction (ELS) questionnaire. The ELS was submitted on the on-line Google-forms, where questions were to be rated on a 5-point Likert scale, with 1 = strongly disagree, 3 = neutral and 5 = strongly agree. E-Learner Satisfaction (ELS) is a reliable and validated questionnaire used to measure students' satisfaction [318, 319] and considers the following questions:

– EL-Q1: This system makes it easy for you to find the content you need.

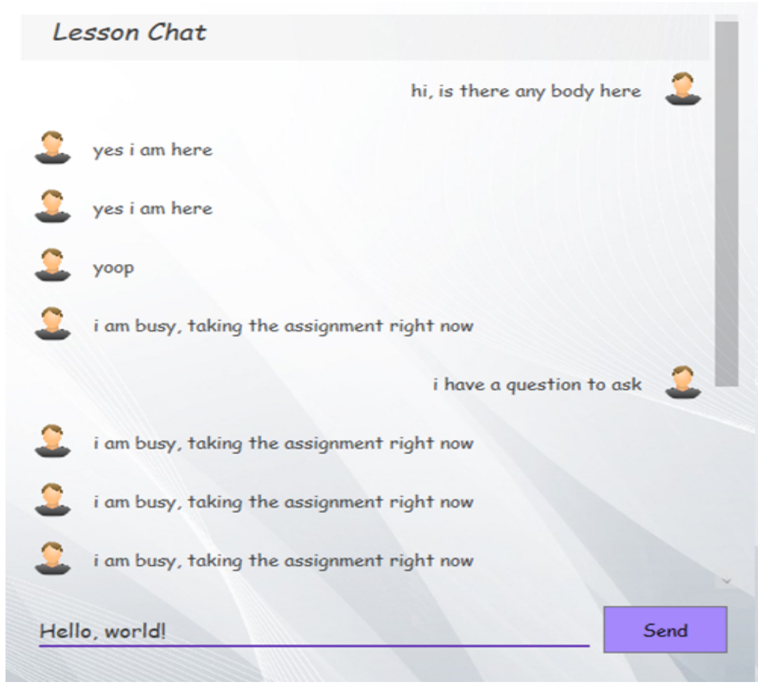


Figure 7.19: Messages sent by students through online chatting

– EL-Q2: This system provides content that exactly fits your needs, and I did not have to look for them in the entire course.

– EL-Q3: This system provides sufficient content.

– EL-Q4: This system makes it easy for you to discuss questions with other students.

– EL-Q5: It was easier to solve my difficulties in understanding the topic with the help of the recommendations of this system.

– EL-Q6: I feel I learn more while using this system.

– EL-Q7: I believe I became productive using this system and its promote my learning interests.

– EL-Q8: I am satisfied with the quality of the personalisation.

– EL-Q9: Overall, I am satisfied with this system.

– EL-Q10: I would recommend this system to my classmates.

The students' satisfaction results are reported in Table 7.4 and it was inferred that 89% of students were satisfied with the ULEARN system and thought that the system's recommended learning objects were very useful.

Table 7.2: Test Cases for Student Portal

Test Case 1 Test Case Name: Student portal				
Test Case ID	**Test Case Scenario**	**Test Case**	**Expected Results**	**Status (Pass /Fail)**
1	Verify login	Validate email and password	Successful login	Pass
2	Verify login	Repeat step 1 for login using false email, password	Un-successful Login should display a warning message "Invalid email or password"	Pass
3	Forget password	Verify the functionality of forget password	Verify after clicked on link, student should navigate to forgot password page.	Pass
4	Forget password	Test login functionality with the newly set password	Successful login	pass
5	New student registration	Enter valid data to register a new student	New student should get created and the system should login user automatically	Pass
6	Log Out	Log out account	Log out redirected to Login page	Pass
7	Learning style questionnaire	Test to see if the learning style questionnaire is loaded when student login.	The expected result is that the first question is loaded on home screen.	Pass
8	welcome screen	Test to see if the ULEARN Welcome Page is loaded on the screen.	The expected result is that the welcome screen with Get personalisation recommendation button is loaded.	Pass
9	Recommendation button	Test to see if personalisation recommendation button can be pressed on in the screen.	The expected result is that the student will get Learning object Recommendation page.	Pass
10	Learning object	Test to see if the student can start learning object.	The expected result is that the learning object button works and leads learning object content on the screen.	Pass
11	Rating button	Test to see if the student can rate learning object.	The expected result is that the student can rate the learning object.	Pass
12	Open Material button	Test if Open Material button can be pressed on learning object screen.	The expected result is that the learning object material open.	Pass
13	Take assignment	Test if the student can start the taking assignment.	The expected result is that the take assignment button works and leads assignment questions on the screen.	Pass

Table 7.3: Networks topics used in the evaluation

Day 1	Day 2	Day 3
-Introduction to Computer Network	- Topologies	-Digital Data Communication Techniques

Table 7.4: Students' satisfaction with the ULEARN system

ELS Questions	Number of Participants									
	Likert Scale									
	Strongly Agree		Agree		Neither		Disagree		Strongly Disagree	
This system makes it easy for you to find the content you need.	35	50%	28	40%	5	7.14%	2	2.86%	0	0%
This system provides content that exactly fits your needs, and I did not have to look for them in the entire course	27	38.58%	33	47.14%	6	8.57%	4	5.71%	0	0%
This system provides sufficient content.	30	42.85%	28	40%	9	12.86%	3	4.29%	0	0%
This system makes it easy for you to discuss questions with other students.	19	27.14%	17	24.29%	14	20%	18	25.71%	2	2.86%
It was easier to solve my difficulties in understanding the topic with the help of the recommendations of this system.	34	48.57%	24	34.28 %	9	12.86%	2	2.86%	1	1.43%
I feel I learn more while using this system.	31	44.28%	27	38.57 %	8	11.43%	3	4.29%	1	1.43%
I believe I became productive using this system and its promote my learning interests.	25	35.70%	37	52.86%	2	2.86%	3	4.29%	3	4.29%
I am satisfied with the quality of the personalisation.	36	51.43%	27	38.57%	3	4.28%	2	2.86%	2	2.86%
Overall, I am satisfied with this system.	34	48.57%	28	40%	5	7.14%	2	2.86%	1	1.43%
I would recommend this system to my classmates.	35	50%	29	41%	4	5.71%	2	2.86%	0	0%

From questions EL-Q1 to EL-Q3, the students were asked about their opinion related to the recommended content by the ULEARN system, as shown in Fig. 7.20. EL-Q1 investigates “whether this system makes it easy for me to find the content you need". Almost 63 out of 70 students were in agreement with this statement whereas just 2 participants disagreed with the rest being neutral.

The purpose behind EL-Q2 was to examine whether the recommended content was exactly fitting students’ preferences. As can be seen in Fig. 7.20, almost 60 students thought that the proposed system’s personalised content matches their preferences, while around 6 students were neutral, and 4 students disagreed.

According to answers for EL-Q3, majority learners noticed that the recommended content was very sufficient to understand the new topic, while only a small number of learners (3 of them) did not have such a feeling.

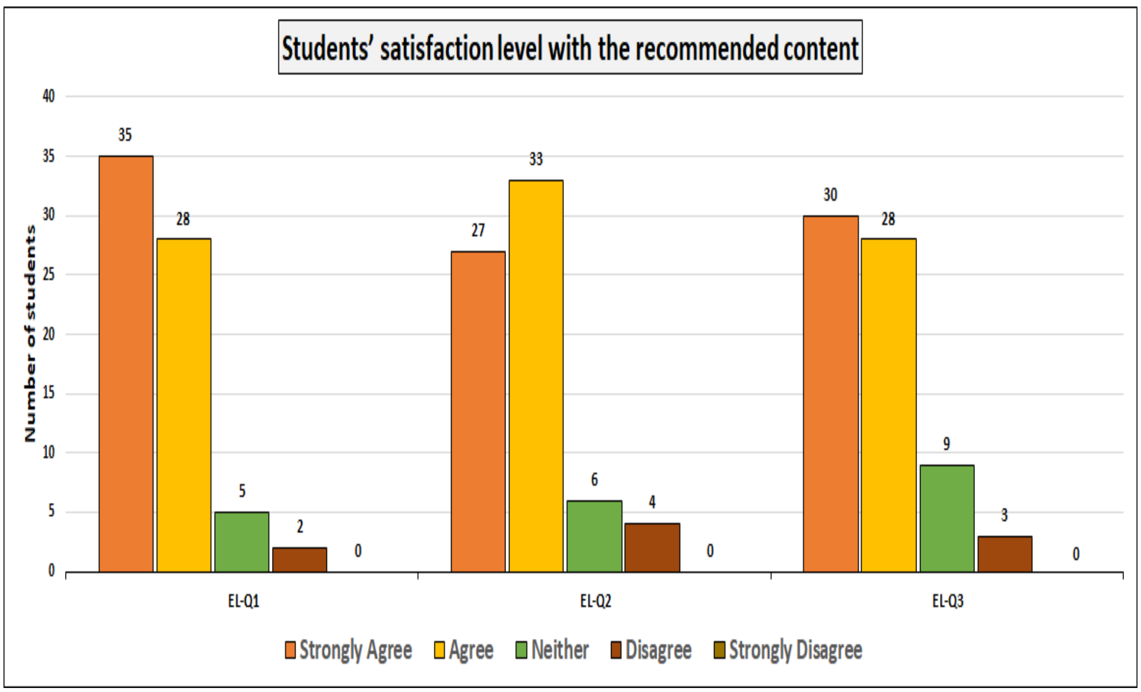


Figure 7.20: Results for questions about assessment satisfaction by the content

Whereas with question EL-Q4, the main intention was to find out from the participants whether this system provides interactive features to discuss with other students. While 19 of them strongly agreed, 17 agreed, and the rest were moderately satisfied, as shown in Fig 7.21. In general, majority of the learners were satisfied with the proposed interactive features while just a minority of students weren’t happy. It was also presumed that many of the unsatisfied answers led to similar recommendations of asking us to allow sharing of learning objects with other students. In future implementations, this option could be enabled for students.

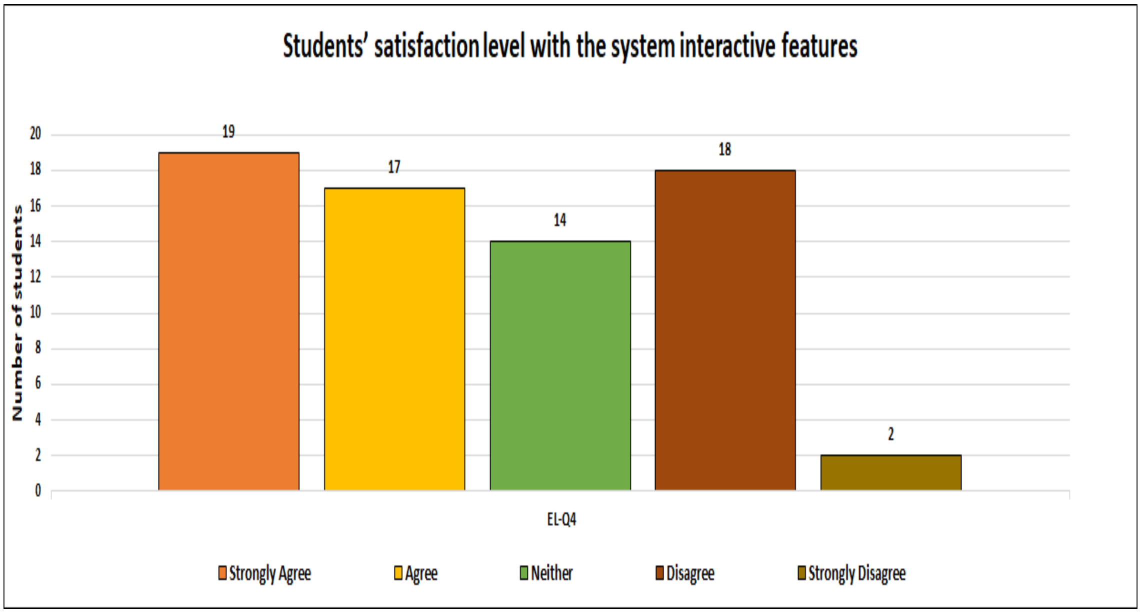


Figure 7.21: Results for questions about assessment satisfaction by the interactive features

From questions EL-Q5 to EL-Q7, we wanted to know whether this system was more helpful in solving any difficulties while understanding a new topic and if it enables participants to be more productive. Fig. 7.22 illustrates the summary of students' responses to these questions. As per EL-Q5, more than 67 out of 70 respondents felt that this system helps them in solving their difficulties while understanding a new topic and enables them to learn what they need.

EL-Q6 asked the learners whether they feel they learn more while using the ULEARN system. Figure 7.22 reveals that more than 58 of them agreed and only 4 students disagreed. According to answers for EL-Q7, participants felt more productive and learned more while using this system where almost 25 of them strongly agreed, 37 agreed and 2 were neutral.

Finally, questions EL-Q8 to EL-Q10 of the questionnaire were designed to measure students' satisfaction with the quality of personalisation in addition to the overall system performance as shown in Fig. 7.23. EL-Q8 attempts to examine whether students are satisfied with the quality of personalisation for which around 63 learners either strongly agreed or agreed, while only 4 of them were in disagreement.

Whereas, EL-Q9 asked learners about their satisfaction and experience while using the ULEARN system. The results showed that about 62 participants were satisfied with their learning experience.

In EL-Q10, we enquired whether students would be happy to recommend this proposed system to

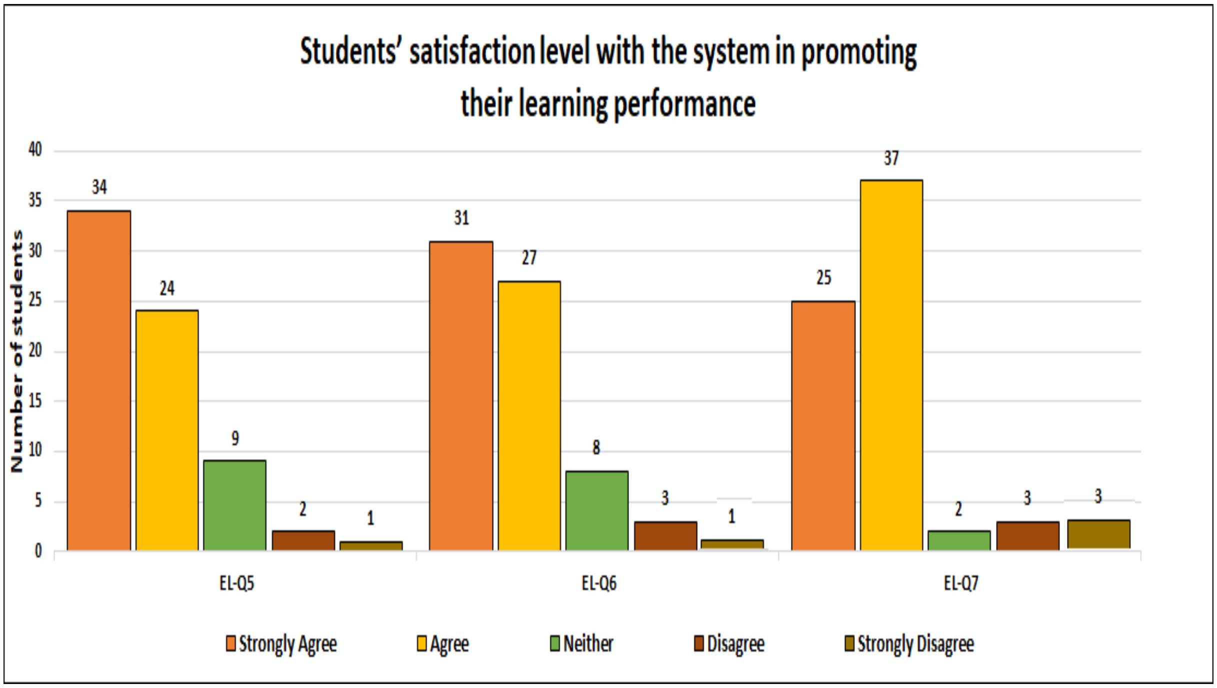


Figure 7.22: Results for questions about measuring the effect of ULEARN system in promoting students learning performance

their colleagues for which 64 learners either strongly agreed or just agreed with no strong negative opinions.

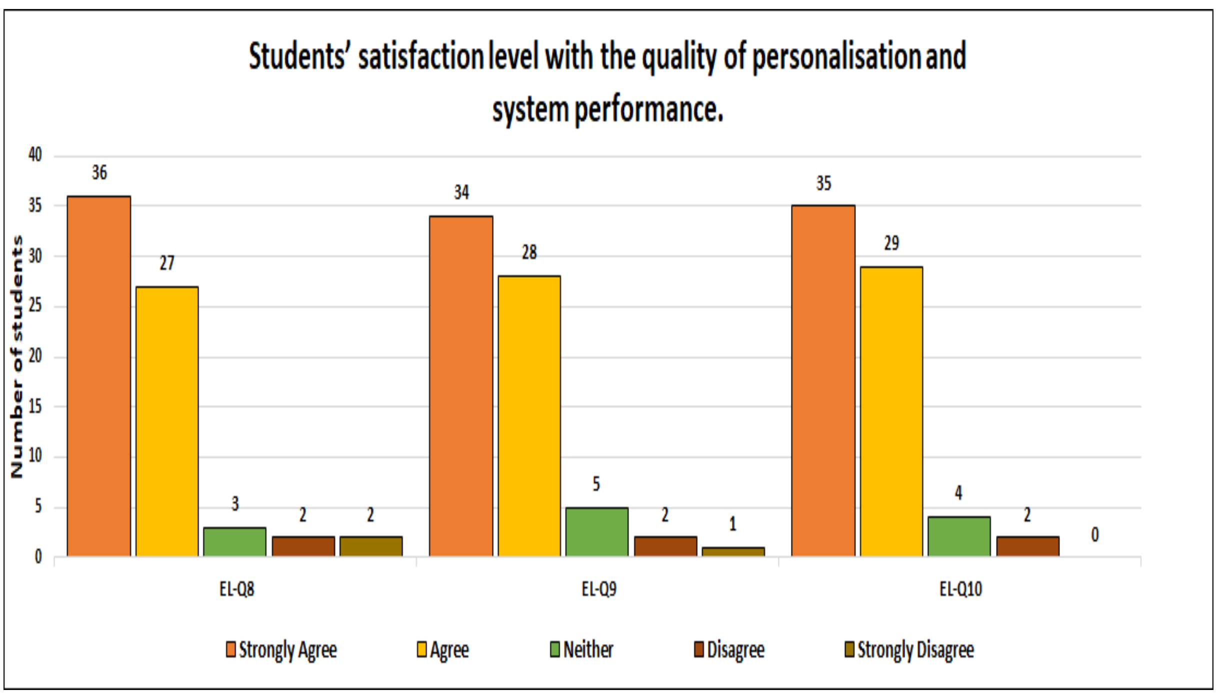


Figure 7.23: Results for questions about assessment satisfaction by the quality of personalisation and system performance.

To conclude with, several crucial findings related to learners' opinions were obtained through this evaluation where it was also inferred a wide number of participants were satisfied with all aspects of personalised learning, specifically with recommended learning objects. Furthermore, the ULEARN system supports students to easily understand recommended content in addition to saving their time searching for suitable materials that match their preferences. Based on the literature, a higher level of satisfaction could lead to more motivated students with increased engagement in the learning process, thus improving their learning outcomes [318, 320, 321]. The following section will discuss the system usability test in detail.

7.3.2 Testing system usability

So as to assess the system's usability, the ULEARN system was evaluated to collect useful information about student experiences by means of an experiment where students were asked to fill-out a System Usability Scale (SUS) post-session questionnaire containing five-point scale Likert statements [322]. The SUS questionnaire was submitted using on-line Google-forms.

This experimental method proves to be a quick and convenient way to assess the usability measure of the system in addition to being a reliable approach of frequently using questionnaires as recommended in the literature [323]. As mentioned previously, the usability questionnaire contained 10 statements for measuring the usability of the ULEARN recommender system where each statement was rated on a 5-point Likert scale ranging from 1 meaning "strongly disagree" to 3 denoting neutrality up till 5 meaning "strongly agree". The statements in the usability questionnaire are as follows:

- SU-S1: I think that I would like to use ULEARN frequently.
- SU-S2: I found the ULEARN system unnecessarily complex.
- SU-S3: I thought the ULEARN system was easy to use.
- SU-S4: I think that I would need the support of a technical person to be able to use this system.
- SU-S5: I found the various functions in this system were well-integrated.
- SU-S6: I thought there was too much inconsistency in this system.
- SU-S7: I would imagine that most students would learn to use this system very quickly.

- SU-S8: I found the ULERAN recommender system very difficult to use.
- SU-S9: I felt very confident using this system.
- SU-S10: I need to learn a lot of things before I could get going with this system.

Moreover, in [324] the author demonstrates that the SUS questionnaire proves to be a valid tool for assessing the LMS usability which is provided to students once they have finished their lessons. To calculate the SUS score, first, the scores from each statement are added together.

For statements SU-S 1, 3, 5, 7 and 9, the score contribution is the scale position minus 1. For statements SU-S 2, 4, 6, 8 and 10, the contribution is 5 minus the scale position. The sum of the scores is then multiplied by 2.5 to obtain the overall value of SUS. Therefore, SUS scores range from 0 to 100, with 100 being the best.

According to the evaluation suggested for SUS results, a score less than 51 corresponds to awful, while 51 to 68 exclusively corresponds to poor, 68 to less than 80.3 exclusively corresponds to good, and a score greater than 80.3 is excellent.

In order to illustrate the way in which the SUS is determined, we have a sample of five test students whose responses to the SUS Questionnaire are shown in Table 7.5. To calculate the SUS score, as follows:

Table 7.5: System usability SUS questionnaire testing sample

Student	**SU-S1**	**SU-S2**	**SU-S3**	**SU-S4**	**SU-S5**	**SU-S6**	**SU-SQ7**	**SU-S8**	**SU-S9**	**SU-S10**
Mary	3	3	4	2	4	3	1	5	4	3
Bob	4	4	5	3	5	1	5	1	3	2
John	4	2	4	1	4	4	3	4	2	3
Tina	5	1	3	4	4	2	4	2	4	2
Rosie	3	2	4	2	5	1	4	3	4	3

- Step 1: Calculate SUS for every student:
  - *Mary* = (3-1) + (5-3) +(4-1) + (5-2) + (4-1) + (5-3) + (1-1)+ (5-5) + (4-1) + (5-3)= 20
  - *Bob* =(4-1) + (5- 4) + (5-1) + (5-3) + (5-1) + (5-1) + (5-1) + (5-1) + (3-1) + (5-2)= 31
  - *John* =(4 -1) + (5-2) + (4-1) + (5-1) + (4-1) + (5-4) + (3-1) + (5-4) + (2-1) + (5- 3)= 30
  - *Tina* = (5-1) + (5-1) + (3-1) + (5-4) + (4-1) + (5-2) + (4-1) + (5-2) + (4-1) + (5-2)= 29
  - *Rosie* = (3-1) + (5-2) + (4-1) + (5- 2)+ (5-1) + (5-1) + (4-1) + (5-3) + (4-1) + (5-3)= 29

– Step 2: Calculate final SUS score:

  * $Mary = 20 \times 2.5 = 50$
  * $Bob = 31 \times 2.5 = 77.5$
  * $John = 30 \times 2.5 = 75$
  * $Tina = 29 \times 2.5 = 72.5$
  * $Rosie = 29 \times 2.5 = 72.5$

  Step 3: Calculate mean to get final SUS score:

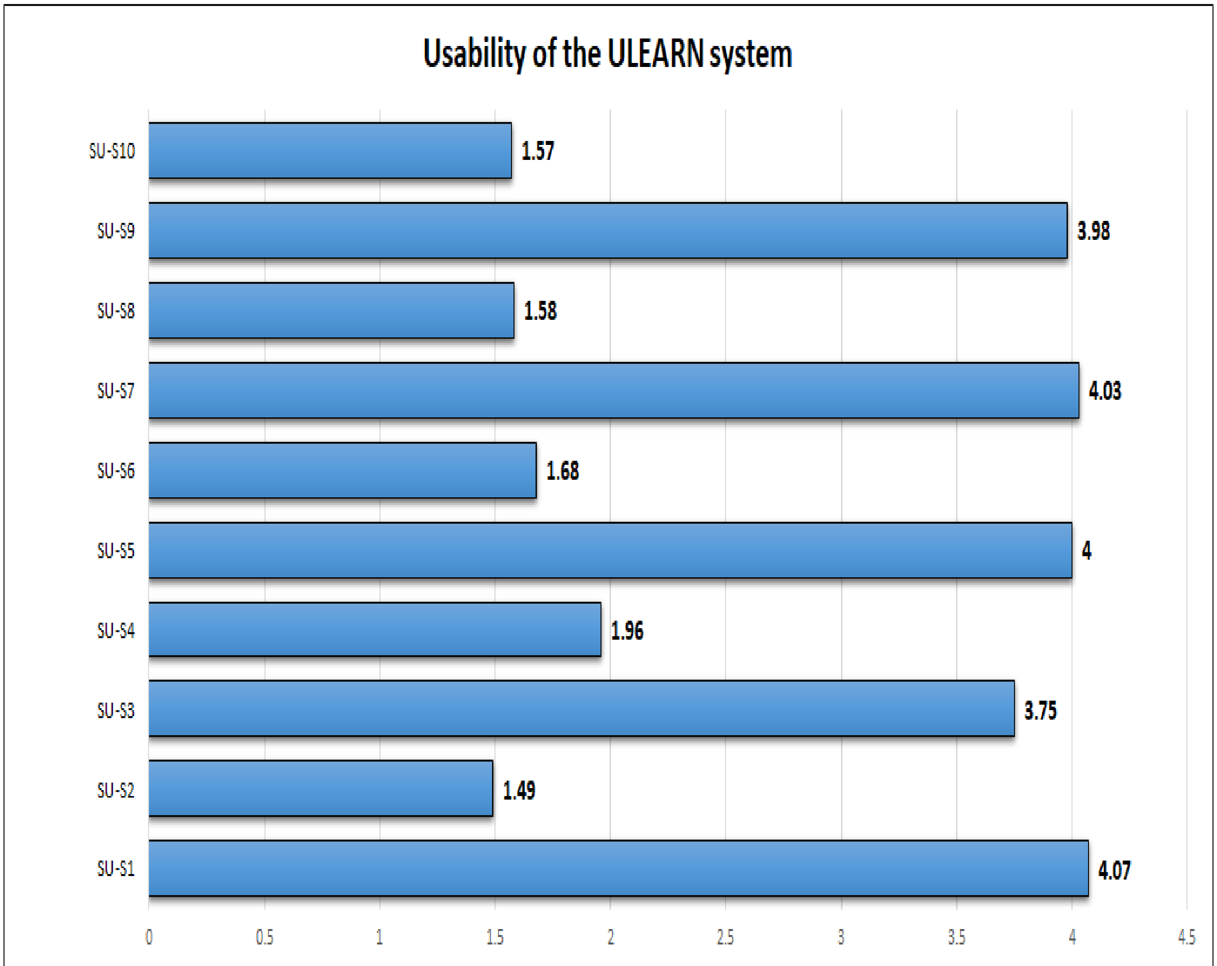
  * SUS Score= $\frac{50+77.5+75+72.5+72.5}{5} = 69.5$

According to Fig. 7.24, students expressed their willingness to repeatedly make use of the system by giving an average score of 4.07 to SU-S1, while a score of SU-S9 was 3.98 reported by a huge number of students in being able to confidently use the system. The SUS score for ULEARN system is 78.9 out of 100 indicating that the ULEARN system is at a 'Good' level of usability. The results also indicate that the participants expressed positive feedback for the overall aspects related to usability in addition to the fact that the majority of students accepted that ULEARN system is usable with respect to learning.

To sum up, on the basis of the results obtained, we can confirm that students perceive high effectiveness, efficiency and satisfaction while making use of the ULEARN system indicating that this system is widely known to be highly usable.

7.4 Summary

In this chapter, the implementation and evaluation of the proposed ULEARN system has been presented. We began with the way in which the design and implementation of the ULEARN recommendation system is carried out on the basis of the architecture discussed in Chapter 3. All ULEARN components are contained in a modular way and can be extended and reused by other e-learning recommendation systems, starting from the initialisation student learning styles, to the recommendation of learning objects up until updating students' profiles on the basis of their interaction with the system. Moreover, the ULEARN system has been evaluated by means of two



	SU-S1	SU-S2	SU-S3	SU-S4	SU-S5	SU-S6	SU-S7	SU-S8	SU-S9	SU-S10
Student responses	4.07	1.49	3.75	1.96	4	1.68	4.03	1.58	3.98	1.57

Figure 7.24: Students responses to the SUS questionnaire

experiments: student satisfaction and, system usability. First, students' satisfaction has been measured through the e-Learner Satisfaction (ELS) questionnaire. The evaluation results demonstrate the efficiency of the ULEARN system since more than 89% of participants were satisfied with the recommendation list where the results met with learning styles. It is also indicated that learners attained a significant level of learning when working with the ULEARN system. Secondly, the system usability has been evaluated through the System Usability Scale (SUS), the results of which have revealed that many students expressed their willingness to use the system again, and that they were quite confident while using the ULEARN system. Based on these findings, the ULEARN system can be considered as a new way of adaptive learning in addition to improving students' learning achievements and performance.

Chapter 8

Conclusions and Recommendations for Future Work

In addition to providing the conclusion to this study, this chapter includes recommendations for future research work within the area of this thesis and is organised as follows. First, Section 8.1 draws up the conclusion of the thesis and summarises the achieved contributions including how proposed e-learning recommendation modules are different from existing work on e-learning recommender systems. Next, Section 8.2 presents the research limitations while future research directions have been outlined for the reader in Section 8.3.

8.1 Conclusion

As part of this research, current literature in the field of this research including recommendation techniques, learning styles models, adaptive e-learning and clustering in the recommendation system, as discussed in Chapter 2 was reviewed. This evaluation of the existing literature was essential in order to identify relevant knowledge gaps in the field of e-learning recommendation systems that need to be addressed as part of this research (Chapter 1, Section 1.2). Certain challenges identified as part of this thesis included current approaches being affected by issues of cold-start and data sparsity, the "one-size-fits-all" approach and low accuracy among others.

In this research, the proposal of a novel e-learning recommender system called ULEARN (Figure 3.2), for the recommendation of personalised learning objects according to students' learning styles was developed. After demonstrating the initialisation of students' learning styles correctly using a dynamic learning styles questionnaire, a sizeable portion of this research covered the way in which the proposed algorithms are capable of overcoming the cold-start and data sparsity problems in addition to improving learning object recommendations. Moreover, this research proved to be successful by the way in which an e-learning recommendation system was developed to dynamically adapt students' learning styles using learning behaviour patterns. This involved the design and development of the ULEARN system for validating the proposed architecture and evaluating the effectiveness of personalised learning objects recommendation from students' learning experiences. A chapter-wise discussion of crucial chapters within this research is presented in the subsequent paragraphs.

Amongst the available recommendation techniques, certain approaches such as Content-Based Filtering, Collaborative Filtering, and Hybrid Filtering were selected as a base for developing the proposed recommender algorithm. Additionally, the Felder-Silverman Learning Style Model (FSLSM) was selected for representing both the students' learning style and learning objects profile through means of a vector of real values ranging from 0 to 1, as described in Section 3.1.

As also mentioned earlier in Chapter 3, the middle layer of the ULEARN recommender system integrates three main modules: Learning Styles initialisation Module (*LS Module*), Learning Objects recommendation Module (*LO Module*), Dynamic Student Profile adaption Module (*DSP Module*). The integration improves the recommendation accuracy and maintains up-to-date learning styles according to students' learning behaviours.

In Chapter 4, we proposed a novel algorithm for initialising student learning styles using a dynamic ILS questionnaire called *LS Module*. First, an empirical study was carried out for the purpose of determining the order of questions for each of the four dimensions of the Felder-Silverman learning style questionnaire. This ranking of questions was then used to build an algorithm that dynamically calculates the initial learning style of users as they go through the questionnaire to just a few questions as they go through the questionnaire, hence saving the user significant time and effort from having to answer all 44 questions of FSLSM questionnaire. As illustrated in the literature review, previous researches in e-learning systems [147,148] are suffering from initialising student preferences at the early stages of recommendation due to the available information not being enough to build users' profiles, which in turn affects the recommendation accuracy (also known as cold-start). Through experimental studies, we have successfully demonstrated the way in which the developed algorithm recognises students' learning style correctly in the initial stage, and how it contributes a new approach to overcome the cold-start problem.

In Chapter 5, we focused intensively on learning object recommendation. A novel *LO Module* was proposed to recommend personalised learning objects according to student learning styles. We have proposed three novel recommendation algorithms (namely, Enhanced Content-Based Filtering (ECBF), Enhanced Collaborative Filtering (ECF) and Enhanced Hybrid Filtering (EHF)). Throughout the course of this research, we concentrated on significantly enhancing the accuracy and quality of recommendations with respect to cold-start and rating sparsity problems. This work is different from other works in the field of learning objects recommendation, as discussed in Chapter 2, current recommendation algorithms suffer from sparsity problem and cold-start problems. They do not take into account the context of LOs while making a recommendation [253, 291, 325]. The proposed algorithms within this piece of research overcome the cold-start problem and rating sparsity by combining learning objects profile (Learning styles) with student profiles along with their ratings while recommending the top-n LOs. Moreover, the proposed algorithms have been tested and evaluated in real-time using real students' datasets, and the results of the experiments determined that the EHF algorithm has the highest recommendation accuracy as compared to ECBF and ECF. The findings also showed that cold-start problems for both new students and new LOs were solved effectively by three algorithms.

In Chapter 6, the *DSP Module* for tracking patterns of student learning behaviour and capturing their learning styles according to the FSLSM was proposed. Through our experiments, we demonstrated the effectiveness of DSP for adapting students' learning styles. In addition, we also showed that the DSP algorithm improves the learning style prediction results in all learning styles dimensions compared to the ILS questionnaire (Chapter 6). The significant difference between our work and literature is that this algorithm detects students learning styles dynamically in addition to maintaining dynamic student profiles within an adaptive e-learning system. In previous researches, [93, 326, 327] developed adaptive educational systems based on static student modelling which was initialised only once when students first used the system.

Finally, we developed a ULEARN system which combined the above three modules in order to validate the proof that integration of dynamic student profiles with the recommendation engine would improve the efficiency and performance of the e-learning recommendation system. Two strategies have been adapted to evaluate ULEARN system. First, we evaluated each module in isolation to examine their performance using student's datasets from AAST-MOODLE log-file. Secondly, we have evaluated all the modules when they are integrated into one system to assess the functionality of this novel system using the user-centred evaluation with real students which allowed us to gain a better understanding of how to provide effective services to users in a real interaction environment (see Chapter 8). In comparison to other works in the literature (e.g. [93, 328, 329]), our proposal provides more insight and a complete evaluation of the various elements of the recommendation system in terms of the learning styles initialisation, student profile adaptation and learning objects recommendations. The results of all evaluations were encouraging, as they provided positive results which support the validity of the proposed models. It was inferred that a majority of students (about 90%) were satisfied with the recommended learning objects, thus indicating that our approach is capable to offer improved recommendation accuracy and, consequently, user satisfaction.

In conclusion, on the basis of the results of this study, the proposed ULEARN system motivates students as well as improves their learning performance. Additionally, the integration of three modules of ULEARN system creates a novel framework which can solve complicated problems such as those of cold-start, data sparsity and static profiles in addition to providing personalised learner experiences and improving the accuracy of recommendations. It is also believed that this research's findings and results will give other researchers a better understanding with respect to the field of this study and aid them in enhancing current adaptive e-learning systems to develop new ones. The following section highlights the limitations of the introduced work.

8.2 Limitations

A specific limitation of this thesis as perceived by the authors was related to the restricted pool of participants of the study who were undergraduate students, mostly from the same university in Egypt, pursuing Information Systems in school of Business Informatics. Therefore, to boost our results, it might be useful to conduct more experiments with new participants from other disciplines, universities or other countries.

Another limitation is that the proposed recommendation algorithm requires a large amount of information to calculate the similarity between students' learning styles. Moreover, there are no public datasets available that contain data about the student behaviour patterns used in ULEARN to dynamically adapt the student learning styles. We believe that in the future the ULEARN system will record more data about students' learning behaviour patterns and build-up its own datasets over time to have a better understanding of learning styles, thus improving learning objects recommendation accuracy. A large number of participants would allow for further large-scale evaluations, so the data gathered and analysed could have a higher degree of statistical significance, and potentially a higher degree of objectivity in measuring the system reliability and scalability. The next section outlines future research directions.

8.3 Future Work

Through the course of this research, there existed potential ideas related to this field of research which were of particular interest to the authors but were not executed within this research since they were not in the current scope. This section briefly discusses possible directions for future study within this area of research.

- First, we plan to combine proposed ECBF, ECF and EHF recommendation algorithms with adaptive learning algorithms such as Bayesian Knowledge Tracing [330], Performance Factors Analysis (PFA) [331], Deep Knowledge Tracing [332], and examine the improved efficiency.
- Another direction of future research is related to enhancement of student profiles through social media. Millions of students are interacting with a large number of social websites in diverse areas through means of videos, photos, communication and entertainment, thus creating a rich and huge resource of information about their preferences and interests. According to [333], such rich social information can benefit current personalisation systems in different ways. We assume that one of these ways is with regards to exploring how such information which might contain students' favourite posts, rating and browsing, can be utilised by ULEARN for the purpose of identifying students' learning styles to enrich their profiles.

- Through this research, we focused on collecting students' feedback through their ratings. As a future direction, ULEARN could interact with the students through smart chat to collect more feedback from them regarding the recommendation of LOs. Moreover, the system could also enhance the quality of recommendation by taking students' feedback while recommending the top-n LOs in the coming sessions.

In addition to the aforementioned future works, a long-term evaluation of the proposed ULEARN system by making use of it in several educational environments would be the ultimate task in the future.

REFERENCES

[1] David Kolb. Experiential learning as the science of learning and development, 1984.

[2] Richard M Felder, Linda K Silverman, et al. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681, 1988.

[3] MS Hasibuan, LE Nugroho, and IP Santosa. Learning style model detection based on prior knowledge in e-learning system. In *2017 Second International Conference on Informatics and Computing (ICIC)*, pages 1–5. IEEE, 2017.

[4] Albert Sangrà, Dimitrios Vlachopoulos, and Nati Cabrera. Building an inclusive definition of e-learning: An approach to the conceptual framework. *The International Review of Research in Open and Distributed Learning*, 13(2):145–159, 2012.

[5] Peter Brusilovsky and Eva Millán. User models for adaptive hypermedia and adaptive educational systems. In *The adaptive web*, pages 3–53. Springer, 2007.

[6] Peter Honey and Alan Mumford. *Learning styles questionnaire*. Organization Design and Development, Incorporated, 1989.

[7] Neil Fleming and C Mills. Vark: A guide to learning styles. 2001.

[8] Vishnu Challam, Susan Gauch, and Aravind Chandramouli. Contextual search using ontology-based user profiles. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, pages 612–617. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2007.

[9] Michele Gorgoglione, Cosimo Palmisano, and Alexander Tuzhilin. Personalization in context: Does context matter when building personalized customer models? In *Sixth International Conference on Data Mining (ICDM'06)*, pages 222–231. IEEE, 2006.

[10] Curtis A Carver, Richard A Howard, and William D Lane. Enhancing student learning through hypermedia courseware and incorporation of student learning styles. *IEEE transactions on Education*, 42(1):33–38, 1999.

[11] Sarah Carr. As distance education comes of age, the challenge is keeping the students. *Chronicle of higher education*, 46(23), 2000.

[12] Fethi A Inan, Erman Yukselturk, and Michael M Grant. Profiling potential dropout students by individual characteristics in an online certificate program. *International Journal of Instructional Media*, 36(2):163–177, 2009.

[13] Sotiris B Kotsiantis, CJ Pierrakeas, and Panayiotis E Pintelas. Preventing student dropout in distance learning using machine learning techniques. In *International conference on knowledge-based and intelligent information and engineering systems*, pages 267–274. Springer, 2003.

[14] Pedro A Willging and Scott D Johnson. Factors that influence students' decision to dropout of online courses. *Journal of Asynchronous Learning Networks*, 13(3):115–127, 2009.

[15] Jaime Teevan, Susan T Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456. ACM, 2005.

[16] Christian Wolf. iweaver: towards learning style-based e-learning in computer science education. In *Proceedings of the fifth Australasian conference on Computing education*, volume 20, pages 273–279. Australian Computer Society, Inc., 2003.

[17] Nikos Manouselis, Hendrik Drachsler, Riina Vuorikari, Hans Hummel, and Rob Koper. Recommender systems in technology enhanced learning. In *Recommender systems handbook*, pages 387–415. Springer, 2011.

[18] Silvia Schiaffino, Patricio Garcia, and Analia Amandi. eteacher: Providing personalized assistance to e-learning students. *Computers and Education*, 51(4):1744–1754, 2008.

[19] Namira Bajraktarevic, Wendy Hall, and Patrick Fullick. Incorporating learning styles in hypermedia environment: Empirical evaluation. In *Proceedings of the workshop on adaptive hypermedia and adaptive web-based systems*, number 1999, pages 41–52, 2003.

[20] Elvira Popescu. Diagnosing students' learning style in an educational hypermedia system. In *Cognitive and emotional processes in Web-based education: Integrating human factors and personalization*, pages 187–208. IGI Global, 2009.

[21] Sabine Graf, Tzu-Chien Liu, and K Kinshuk. Interactions between students learning styles, achievement and behaviour in mismatched courses. In *Proceedings of the international conference on cognition and exploratory learning in digital age (CELDA 2008). IADIS International Conference*, pages 223–230. Citeseer, 2008.

[22] Jens O Liegle and Thomas N Janicki. The effect of learning styles on the navigation needs of web-based learners. *Computers in human behavior*, 22(5):885–898, 2006.

[23] Enrique Alfonseca, Rosa M Carro, Estefanía Martín, Alvaro Ortigosa, and Pedro Paredes. The impact of learning styles on student grouping for collaborative learning: a case study. *User Modeling and User-Adapted Interaction*, 16(3-4):377–401, 2006.

[24] Sabine Graf, Chung Hsien Lan, Tzu-Chien Liu, et al. Investigations about the effects and effectiveness of adaptivity for students with different learning styles. In *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on*, pages 415–419. IEEE, 2009.

[25] W Andrew Taylor. Computer-mediated knowledge sharing and individual user differences: an exploratory study. *European journal of information systems*, 13(1):52–64, 2004.

[26] Sabine Graf et al. Using cognitive traits for improving the detection of learning styles. In *Database and Expert Systems Applications (DEXA), 2010 Workshop on*, pages 74–78. IEEE, 2010.

[27] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[28] Rui Ren, Lingling Zhang, Limeng Cui, Bo Deng, and Yong Shi. Personalized financial news recommendation algorithm based on ontology. *Procedia Computer Science*, 55:843–851, 2015.

[29] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, (6):734–749, 2005.

[30] Inma Garcia, Laura Sebastia, and Eva Onaindia. On the design of individual and group recommender systems for tourism. *Expert systems with applications*, 38(6):7683–7692, 2011.

[31] Alan F Smeaton and Jamie Callan. Personalisation and recommender systems in digital libraries. *International Journal on digital libraries*, 5(4):299–308, 2005.

[32] Jae Kyeong Kim, Yoon Ho Cho, Woo Ju Kim, Je Ran Kim, and Ji Hae Suh. A personalized recommendation procedure for internet shopping support. *Electronic commerce research and applications*, 1(3-4):301–313, 2002.

[33] Jian Wang, Badrul Sarwar, and Neel Sundaresan. Utilizing related products for post-purchase recommendation in e-commerce. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 329–332. ACM, 2011.

[34] Iván Cantador. Exploiting the conceptual space in hybrid recommender systems: a semantic-based approach. 2008.

[35] Daniel Billsus and Michael J Pazzani. User modeling for adaptive news access. *User modeling and user-adapted interaction*, 10(2-3):147–180, 2000.

[36] Ting-Peng Liang and Hung-Jen Lai. Discovering user interests from web browsing behavior: An application to internet news services. In *Proceedings of the 35th annual Hawaii international conference on system sciences*, pages 2718–2727. IEEE, 2002.

[37] Ben Horsburgh, Susan Craw, and Stewart Massie. Learning pseudo-tags to augment sparse tagging in hybrid music recommender systems. *Artificial Intelligence*, 219:25–39, 2015.

[38] Jose Aguilar, Priscila Valdiviezo-Díaz, and Guido Riofrio. A general framework for intelligent recommender systems. *Applied Computing and Informatics*, 13(2):147–160, 2017.

[39] Muhammad Waseem Chughtai, Ali Selamat, Imran Ghani, and Jason J Jung. Retracted: E-learning recommender systems based on goal-based hybrid filtering. *International Journal of Distributed Sensor Networks*, 10(7):912130, 2014.

[40] Hyun Jin Cha, Yong Se Kim, Seon Hee Park, Tae Bok Yoon, Young Mo Jung, and Jee-Hyong Lee. Learning styles diagnosis based on user interface behaviors for the customization of learning interfaces in an intelligent tutoring system. In *International Conference on Intelligent Tutoring Systems*, pages 513–524. Springer, 2006.

[41] Ömer Şimşek, Nilüfer Atman, Mustafa Murat İnceoğlu, and Yüksel Deniz Arikan. Diagnosis of learning styles based on active/reflective dimension of felder and silverman's learning style model in a learning management system. In *International Conference on Computational Science and Its Applications*, pages 544–555. Springer, 2010.

[42] Nuanwan Soonthornphisaj, Ekkawut Rojsattarat, and Sukanya Yim-Ngam. Smart e-learning using recommender system. In *International conference on intelligent computing*, pages 518–523. Springer, 2006.

[43] Jorge Bozo, Rosa Alarcón, and Sebastian Iribarra. Recommending learning objects according to a teachers' contex model. In *European Conference on Technology Enhanced Learning*, pages 470–475. Springer, 2010.

[44] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.

[45] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[46] Mengsi Liu, Weike Pan, Miao Liu, Yaofeng Chen, Xiaogang Peng, and Zhong Ming. Mixed similarity learning for recommendation with implicit feedback. *Knowledge-Based Systems*, 119:178–185, 2017.

[47] Tulasi K Paradarami, Nathaniel D Bastian, and Jennifer L Wightman. A hybrid recommender system using artificial neural networks. *Expert Systems with Applications*, 83:300–313, 2017.

[48] Sunita B Aher and LMRJ Lobo. Combination of machine learning algorithms for recommendation of courses in e-learning system based on historical data. *Knowledge-Based Systems*, 51:1–14, 2013.

[49] Ig Ibert Bittencourt, Seiji Isotani, Evandro Costa, and Riichiro Mizoguchi. Research directions on semantic web and education. *Interdisciplinary Studies in Computer Science*, 19(1):60–67, 2008.

[50] Maria Dominic and Sagayaraj Francis. An adaptable e-learning architecture based on learners' profiling. *IJ Modern Education and Computer Science*, 3:26–31, 2015.

[51] Gladys Castillo, João Gama, and Ana M Breda. An adaptive predictive model for student modeling. In *Advances in Web-based education: Personalized learning environments*, pages 70–92. IGI Global, 2006.

[52] Sabine Graf, Tzu-Chien Liu, et al. Identifying learning styles in learning management systems by using indications from students' behaviour. In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference*, pages 482–486. IEEE, 2008.

[53] Patricio García, Analía Amandi, Silvia Schiaffino, and Marcelo Campo. Evaluating bayesian networks' precision for detecting students' learning styles. *Computers & Education*, 49(3):794–808, 2007.

[54] Sabine Graf and Prof Kinshuk. An approach for detecting learning styles in learning management systems. In *Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06)*, pages 161–163. IEEE, 2006.

[55] Gordana Dodig Crnkovic. Constructive research and info-computational knowledge generation. In *Model-Based Reasoning in Science and Technology*, pages 359–380. Springer, 2010.

[56] Eva Labro and Tero-Seppo Tuomela. On bringing more action into management accounting research: process considerations based on two constructive case studies. *European Accounting Review*, 12(3):409–442, 2003.

[57] Dragos Simandan. Introduction: Learning as a geographical process. *The Professional Geographer*, 65(3):363–368, 2013.

[58] Peggy A Ertmer and Timothy J Newby. Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective. *Performance improvement quarterly*, 26(2):43–71, 2013.

[59] P Karen Murphy and Stephanie L Knight. Exploring a century of advancements in the science of learning. *Review of Research in Education*, 40(1):402–456, 2016.

[60] George Graham. Behaviorism. 2000.

[61] Peggy A Ertmer and Timothy J Newby. Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective. *Performance improvement quarterly*, 6(4):50–72, 1993.

[62] Marcy Perkins Driscoll. *Psychology of learning for instruction.* Allyn & Bacon, 1994.

[63] Fergus IM Craik and Endel Tulving. Depth of processing and the retention of words in episodic memory. *Journal of experimental Psychology: general*, 104(3):268, 1975.

[64] Fergus IM Craik and Robert S Lockhart. Levels of processing: A framework for memory research. *Journal of verbal learning and verbal behavior*, 11(6):671–684, 1972.

[65] Buthaina Deeb, Zainuddin Hassan, and Majdi Beseiso. An adaptive hmm based approach for improving e-learning methods. In *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, pages 1–7. IEEE, 2014.

[66] Huong May Truong. Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities. *Computers in human behavior*, 55:1185–1193, 2016.

[67] Sucheta V Kolekar, SG Sanjeevi, and DS Bormane. Learning style recognition using artificial neural network for adaptive user interface in e-learning. In *2010 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–5. IEEE, 2010.

[68] Mohammad Alshammari. *Adaptation based on learning style and knowledge level in e-learning systems*. PhD thesis, University of Birmingham, 2016.

[69] Khalid Almohammadi and Hani Hagras. An interval type-2 fuzzy logic based system for customised knowledge delivery within pervasive e-learning platforms. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2872–2879. IEEE, 2013.

[70] Giuseppe Fenza, Francesco Orciuoli, and Demetrios G Sampson. Building adaptive tutoring model using artificial neural networks and reinforcement learning. In *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*, pages 460–462. IEEE, 2017.

[71] Ankit Malpani, Balaraman Ravindran, and Hema Murthy. Personalized intelligent tutoring system using reinforcement learning. In *Twenty-Fourth International FLAIRS Conference*, 2011.

[72] Chun-Hui Wu, Ta-Cheng Chen, Yih-Her Yan, and Chun-Feng Lee. Developing an adaptive e-learning system for learning excel. In *2017 International Conference on Applied System Innovation (ICASI)*, pages 1973–1975. IEEE, 2017.

[73] Jacob Whitehill and Javier Movellan. Approximately optimal teaching of approximately optimal learners. *IEEE Transactions on Learning Technologies*, 11(2):152–164, 2017.

[74] Ramkumar Rajendran, Sridhar Iyer, and Sahana Murthy. Personalized affective feedback to address students' frustration in its. *IEEE Transactions on Learning Technologies*, 12(1):87–97, 2018.

[75] LB Krithika. Student emotion recognition system (sers) for e-learning improvement based on learner concentration metric. *Procedia Computer Science*, 85:767–776, 2016.

[76] Pierre Chalfoun, Soumaya Chaffar, and Claude Frasson. Predicting the emotional reaction of the learner with a machine learning technique. In *Workshop on Motivaional and Affective Issues in ITS, ITS'06, International Conference on Intelligent Tutoring Systems*, 2006.

[77] Cristina Conati and Xiaoming Zhou. Modeling students' emotions from cognitive appraisal in educational games. In *International Conference on Intelligent Tutoring Systems*, pages 944–954. Springer, 2002.

[78] Somayeh Fatahi, Mohammad Kazemifard, and Nasser Ghasem-Aghaee. Design and implementation of an e-learning model by considering learner's personality and emotions. In *Advances in electrical engineering and computational science*, pages 423–434. Springer, 2009.

[79] Jennifer Sabourin, Bradford Mott, and James C Lester. Modeling learner affect with theoretically grounded dynamic bayesian networks. In *International Conference on Affective Computing and Intelligent Interaction*, pages 286–295. Springer, 2011.

[80] Sidney K D'mello, Scotty D Craig, Amy Witherspoon, Bethany Mcdaniel, and Arthur Graesser. Automatic detection of learner's affect from conversational cues. *User modeling and user-adapted interaction*, 18(1-2):45–80, 2008.

[81] Cristina Conati and Heather Maclaren. Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, 19(3):267–303, 2009.

[82] Chih-Ming Chen and Yi-Lun Li. Personalised context-aware ubiquitous learning system for supporting effective english vocabulary learning. *Interactive Learning Environments*, 18(4):341–364, 2010.

[83] Isabela Gasparini, Marcelo S Pimenta, and José Palazzo M de Oliveira. How to apply context-awareness in an adaptive e-learning environment to improve personalization capabilities? In *2011 30th International Conference of the Chilean Computer Science Society*, pages 161–170. IEEE, 2011.

[84] Vicki Jones and Jun H Jo. Ubiquitous learning environment: An adaptive teaching system using ubiquitous technology. In *Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference*, volume 468, page 474. Perth, Western Australia, 2004.

[85] Chong Woo Woo, Martha W Evens, Reva Freedman, Michael Glass, Leem Seop Shim, Yuemei Zhang, Yujian Zhou, and Joel Michael. An intelligent tutoring system that generates a natural language dialogue using dynamic multi-level planning. *Artificial intelligence in medicine*, 38(1):25–46, 2006.

[86] Milan Gnjatović. Conversational agents and negative lessons from behaviourism. In *Innovations in Big Data Mining and Embedded Knowledge*, pages 259–274. Springer, 2019.

[87] Arzu Gul. *Constructivism as a New Notion in English Language Education in Turkey*. PhD thesis, Kent State University, 2016.

[88] Carol Evans. L.-f. zhang: The malleability of intellectual styles, 2015.

[89] Maria Kozhevnikov, Carol Evans, and Stephen M Kosslyn. Cognitive style as environmentally sensitive individual differences in cognition: A modern synthesis and applications in education, business, and management. *Psychological Science in the Public Interest*, 15(1):3–33, 2014.

[90] Richard M Felder. Matters of style. *ASEE prism*, 6(4):18–23, 1996.

[91] James W Keefe. *Learning Style Theory and Practice.* ERIC, 1987.

[92] Alice Y Kolb and David A Kolb. Learning styles and learning spaces: A review of the multidisciplinary application of experiential learning theory in higher education. In *Learning styles and learning: A key to meeting the accountability demands in education*, pages 45–91. Nova Science Publishers New York, 2006.

[93] Sabine Graf. Adaptivity in learning management systems focussing on learning styles, 2007.

[94] Isabel Briggs Myers. The myers-briggs type indicator: Manual (1962). 1962.

[95] Robert Loo. Confirmatory factor analyses of kolb's learning style inventory (lsi-1985). *British Journal of Educational Psychology*, 69(2):213–219, 1999.

[96] Hanna Schröder, Alexandra Henke, Lina Stieger, Stefan Beckers, Henning Biermann, Rolf Rossaint, and Saša Sopka. Influence of learning styles on the practical performance after the four-step basic life support training approach–an observational cohort study. *PloS one*, 12(5):e0178210, 2017.

[97] Abdullah Konak, Tricia K Clark, and Mahdi Nasereddin. Using kolb's experiential learning cycle to improve student learning in virtual computer laboratories. *Computers & Education*, 72:11–22, 2014.

[98] Jasim Mohamed Hasan Yousif Almarzooqi. *An evaluation of the effectiveness of face-to-face versus e-learning in the UAE Civil Defence sector*. PhD thesis, University of the West of England, 2019.

[99] Peter Honey, Alan Mumford, et al. The manual of learning styles. 1992.

[100] Christopher W Allinson and John Hayes. Validity of the learning styles questionnaire. *Psychological Reports*, 67(3):859–866, 1990.

[101] Carl Gustav Jung. 1971. psychological types. *Collected Works of CG Jung*, 6, 1921.

[102] Frank Coffield, David Moseley, Elaine Hall, Kathryn Ecclestone, et al. Learning styles and pedagogy in post-16 learning: A systematic and critical review, 2004.

[103] Daniel W Salter, Nancy J Evans, and Deanna S Forney. A longitudinal study of learning style preferences on the myers-briggs type indicator and learning style inventory. *Journal of College Student Development*, 47(2):173–184, 2006.

[104] Stavros K Filippidis and Ioannis A Tsoukalas. On the use of adaptive instructional images based on the sequential–global dimension of the felder–silverman learning style theory. *Interactive Learning Environments*, 17(2):135–150, 2009.

[105] John A Clarke. Cognitive style and computer-assisted learning: problems and a possible solution1. *Alt-j*, 1(1):47–59, 1993.

[106] Jasna Kuljis and Fang Liu. A comparison of learning style theories on the suitability for elearning. *Proceedings of the IASTED Conference on Web Technologies, Applications, and Services*, 2005:191–197, 2005.

[107] Yavuz Akbulut and Cigdem Suzan Cardak. Adaptive educational hypermedia accommodating learning styles: A content analysis of publications from 2000 to 2011. *Computers & Education*, 58(2):835–842, 2012.

[108] Yong-Ming Huang. Exploring the factors that affect the intention to use collaborative technologies: The differing perspectives of sequential/global learners. *Australasian Journal of Educational Technology*, 31(3), 2015.

[109] Richard M Felder and Joni Spurlin. Applications, reliability and validity of the index of learning styles. *International journal of engineering education*, 21(1):103–112, 2005.

[110] Thomas A Litzinger, Sang Ha Lee, and John C Wise. A study of the reliability and validity of the felder-soloman index of learning styles. In *Proceedings of the 2005 American Society for Education Annual Conference & Exposition*, pages 1–16, 2005.

[111] Nigel Van Zwanenberg, LJ Wilkinson, and A Anderson. Felder and silverman's index of learning styles and honey and mumford's learning styles questionnaire: how do they compare and do they predict academic performance? *Educational Psychology*, 20(3):365–380, 2000.

[112] Malgorzata S Zywno. A contribution of validation of score meaning for felder-soloman's. In *Index of Learning Styles, Proc. 2003 Annual ASEE Conference, ASEE*. Citeseer, 2003.

[113] Elizabeth Brown, Tony Fisher, and Tim Brailsford. Real users, real results: examining the limitations of learning styles within aeh. In *Proceedings of the eighteenth conference on Hypertext and hypermedia*, pages 57–66. ACM, 2007.

[114] Funda Dağ and Aynur Geçer. Relations between online learning and learning styles. *Procedia-Social and Behavioral Sciences*, 1(1):862–871, 2009.

[115] Chia-Cheng Hsu, Kun-Te Wang, and Yueh-Min Huang. Modeling personalized learning styles in a web-based learning system. In *Transactions on edutainment IV*, pages 12–21. Springer, 2010.

[116] Elvira Popescu. Adaptation provisioning with respect to learning styles in a web-based educational system: an experimental study. *Journal of Computer Assisted Learning*, 26(4):243–257, 2010.

[117] Curtis A Carver Jr, Richard A Howard, and Edward Lavelle. Enhancing student learning by incorporating learning styles into adaptive hypermedia. In *EdMedia+ Innovate Learning*, pages 28–32. Association for the Advancement of Computing in Education (AACE), 1996.

[118] Clara-Inés Peña, Jose-L Marzo, and Josep-Lluis de la Rosa. Intelligent agents in a teaching and learning environment on the web. In *Proceedings of the international conference on advanced learning technologies*, pages 21–27. IEEE Learning Technology Task Force. Palmerston North, NZ, 2002.

[119] Pedro Paredes and Pilar Rodriguez. A mixed approach to modelling learning styles in adaptive educational hypermedia. *Advanced Technology for Learning*, 1(4):210–215, 2004.

[120] Norazlina Ahmad and Zaidatun Tasir. Threshold value in automatic learning style detection. *Procedia-Social and Behavioral Sciences*, 97:346–352, 2013.

[121] Juan Feldman, Ariel Monteserin, and Analía Amandi. Automatic detection of learning styles: state of the art. *Artificial Intelligence Review*, 44(2):157–186, 2015.

[122] Ebru Özpolat and Gözde B Akar. Automatic detection of learning styles for an e-learning system. *Computers and Education*, 53(2):355–367, 2009.

[123] MS Hasibuan and LE Nugroho. Detecting learning style using hybrid model. In *2016 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, pages 107–111. IEEE, 2016.

[124] Magdalini Eirinaki, Dimitrios Mavroeidis, George Tsatsaronis, and Michalis Vazirgiannis. Introducing semantics in web personalization: The role of ontologies. In *Semantics, web and mining*, pages 147–162. Springer, 2005.

[125] Bamshad Mobasher and Sarabjot Singh Anand. *Intelligent Techniques for Web Personalization: IJCAI 2003 Workshop, ITWP 2003, Acapulco, Mexico, August 11, 2003, Revised Selected Papers*, volume 3169. Springer, 2005.

[126] Kyparisia A Papanikolaou, Maria Grigoriadou, Harry Kornilakis, and George D Magoulas. Personalizing the interaction in a web-based educational hypermedia system: the case of inspire. *User modeling and user-adapted interaction*, 13(3):213–267, 2003.

[127] Annabel Latham, Keeley Crockett, and David Mclean. Profiling student learning styles with multilayer perceptron neural networks. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2510–2515. IEEE, 2013.

[128] Fabiano A Dorça, Luciano V Lima, Márcia A Fernandes, and Carlos R Lopes. Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: An experimental analysis. *Expert Systems with Applications*, 40(6):2092–2101, 2013.

[129] Esperance Mwamikazi, Philippe Fournier-Viger, Chadia Moghrabi, Adnen Barhoumi, and Robert Baudouin. An adaptive questionnaire for automatic identification of learning styles. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 399–409. Springer, 2014.

[130] Alvaro Ortigosa, Pedro Paredes, and Pilar Rodriguez. Ah-questionnaire: An adaptive hierarchical questionnaire for learning styles. *Computers & Education*, 54(4):999–1005, 2010.

[131] Victoria Tsiriga and Maria Virvou. Initializing student models in web-based itss: a generic approach. In *Proceedings 3rd IEEE International Conference on Advanced Technologies*, pages 42–46. IEEE, 2003.

[132] I Beaumont. User modelling and hypertext adaptation in the tutoring system anatom-tutor. In *UM'94 Fourth International Conference on User Modelling*, volume 4, pages 21–45, 1994.

[133] E. S. Amir, M. Sumadyo, D. I. Sensuse, Y. G. Sucahyo, and H. B. Santoso. Automatic detection of learning styles in learning management system by using literature-based method and support vector machine. pages (41–144). IEEE Press, 2016.

[134] Enrique Frias-Martinez, G Magoulas, S Chen, and R Macredie. Modeling human behavior in user-adaptive systems: Recent advances using soft computing techniques. *Expert Systems with Applications*, 29(2):320–329, 2005.

[135] Huong May Truong. Integrating learning styles into adaptive e-learning system. *International Educational Data Mining Society*, 2015.

[136] Amir Albadvi and Mohammad Shahbazi. A hybrid recommendation technique based on product category attributes. *Expert Systems with Applications*, 36(9):11480–11488, 2009.

[137] Sotiris Botsios, Demetrius Georgiou, and Nikolaos Safouris. Contributions to adaptive educational hypermedia systems via on-line learning style estimation. *Journal of Educational Technology & Society*, 11(2):322–339, 2008.

[138] Patricio García, Analía Amandi, Silvia Schiaffino, and Marcelo Campo. Using bayesian networks to detect students' learning styles in a web-based education system. *Proc of ASAI, Rosario*, pages 115–126, 2005.

[139] Virginia Yannibelli, Daniela Godoy, and Analía Amandi. A genetic algorithm approach to recognise students' learning styles. *Interactive Learning Environments*, 14(1):55–78, 2006.

[140] Sabine Graf and S Viola. Automatic student modelling for detecting learning style preferences in learning management systems. In *Proc. international conference on cognition and exploratory learning in digital age*, pages 172–179, 2009.

[141] Norazlina Ahmad, Zaidatun Tasir, Jamri Kasim, and Harun Sahat. Automatic detection of learning styles in learning management systems by using literature-based method. *Procedia-Social and Behavioral Sciences*, 103:181–189, 2013.

[142] Pham Quang Dung and Adina Magda Florea. A literature-based method to automatically detect learning styles in learning management systems. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, page 46. ACM, 2012.

[143] Nilüfer Atman, Mustafa Murat Inceoğlu, and Burak Galip Aslan. Learning styles diagnosis based on learner behaviors in web based learning. In *International Conference on Computational Science and Its Applications*, pages 900–909. Springer, 2009.

[144] Thair Hamtini and Hadeel Ateia. A proposed dynamic technique for detecting learning style using literature based approach. In *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–6. IEEE, 2015.

[145] M Prabhani Pitigala Liyanage, KS Lasith Gunawardena, and Masahito Hirakawa. A framework for adaptive learning management systems using learning styles. In *2013 International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 261–265. IEEE, 2013.

[146] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[147] Aldo Ramirez-Arellano, Juan Bory-Reyes, and Luis Manuel Hernández-Simón. Learning object assembly based on learning styles. In *Smart Education and e-Learning 2016*, pages 447–462. Springer, 2016.

[148] Kamal Souali, Abdellatif El Afia, Rdouan Faizi, and Raddouane Chiheb. A new recommender system for e-learning environments. In *2011 International Conference on Multimedia Computing and Systems*, pages 1–4. IEEE, 2011.

[149] Elvira Popescu, Costin Badica, and Lucian Moraret. Accommodating learning styles in an adaptive educational system. *Informatica*, 34(4), 2010.

[150] Silvia Baldiris, Ramón Fabregat, Carolina Mejía, and Sergio Gómez. Adaptation decisions and profiles exchange among open learning management systems based on agent negotiations and machine learning techniques. In *International Conference on Human-Computer Interaction*, pages 12–20. Springer, 2009.

[151] Hazra Imran, Mohammad Belghis-Zadeh, Ting-Wen Chang, Sabine Graf, et al. Plors: a personalized learning object recommender system. *Vietnam Journal of Computer Science*, 3(1):3–13, 2016.

[152] Aleksandra Klašnja-Milićević, Boban Vesin, Mirjana Ivanović, and Zoran Budimac. E-learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education*, 56(3):885–899, 2011.

[153] Annabel Latham, Keeley Crockett, David McLean, and Bruce Edmonds. A conversational intelligent tutoring system to automatically predict learning styles. *Computers and Education*, 59(1):95–109, 2012.

[154] Evangelos Triantafillou, Andreas Pomportsis, and Elissavet Georgiadou. Aes-cs: adaptive educational system based on cognitive styles. In *Second International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Malaga, Spain*, 2002.

[155] El Hassan Abelwahed Essaid El Bachari and Mohammed El Adnani. E-learning personalization based on dynamic learners'preference. 2011.

[156] Paul De Bra and Licia Calvi. Aha: a generic adaptive hypermedia system. In *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia*, pages 5–12. sn, 1998.

[157] Paul Arabasz and Mary Beth Baker. Evolving campus support models for e-learning courses. *Educause Center for Applied Research Bulletin*, 1(9):9, 2003.

[158] Shu-Sheng Liaw, Hsiu-Mei Huang, and Gwo-Dong Chen. Surveying instructor and learner attitudes toward e-learning. *Computers & Education*, 49(4):1066–1080, 2007.

[159] Robert A Reiser. A history of instructional design and technology: Part ii: A history of instructional design. *Educational technology research and development*, 49(2):57–67, 2001.

[160] Dianne Conrad. Deep in the hearts of learners: Insights into the nature of online community. *International Journal of E-Learning & Distance Education/Revue internationale du e-learning et la formation à distance*, 17(1):1–19, 2002.

[161] Beverley E Crane. *Online Teaching and Learning: A Practical Guide for Librarians*, volume 29. Rowman & Littlefield, 2016.

[162] Julia Duggleby. *How to be an online tutor*. Gower Publishing, Ltd., 2000.

[163] Sergio Bermejo. Cooperative electronic learning in virtual laboratories through forums. *IEEE Transactions on Education*, 48(1):140–149, 2005.

[164] Alex Koohang and Keith Harman. Open source: A metaphor for e-learning. *Informing Science*, 8, 2005.

[165] Taerim Lee and Jungjin Lee. Quality assurance of web based e-learning for statistical education. In *Compstat 2006-Proceedings in Computational Statistics*, pages 429–438. Springer, 2006.

[166] Fernando Alonso, Genoveva López, Daniel Manrique, and José M Viñes. An instructional model for web-based e-learning education with a blended learning process approach. *British Journal of educational technology*, 36(2):217–235, 2005.

[167] Clark Aldrich. *Simulations and the future of learning: An innovative (and perhaps revolutionary) approach to e-learning*. John Wiley & Sons, 2003.

[168] Andrea Hope and Patrick Guiton. *Strategies for sustainable open and distance learning*, volume 6. Psychology Press, 2006.

[169] Joi L Moore, Camille Dickson-Deane, and Krista Galyen. e-learning, online learning, and distance learning environments: Are they the same? *The Internet and Higher Education*, 14(2):129–135, 2011.

[170] Tracey Wilen-Daugenti. *edu: Technology and learning environments in higher education*. Peter Lang, 2009.

[171] Paul Trafford and Yukari Shirota. An introduction to virtual learning environments. *Gakushuin Econ. Pap*, 48(10):143–151, 2011.

[172] Răzvan Rughiniş, Bogdan Cristian Doinea, Cristian Săndescu, and Elena Iancu. The we-learn distributed content management system. In *9th RoEduNet IEEE International Conference*, pages 123–127. IEEE, 2010.

[173] Morten Flate Paulsen. Online education systems: Discussion and definition of terms. *NKI distance education*, 202, 2002.

[174] Barbara Hayes-Roth. A blackboard architecture for control. *Artificial intelligence*, 26(3):251–321, 1985.

[175] Martin Dougiamas and Peter Taylor. Moodle: Using learning communities to create an open source course management system. In *EdMedia+ Innovate Learning*, pages 171–178. Association for the Advancement of Computing in Education (AACE), 2003.

[176] Nadire Cavus and Teyang Zabadi. A comparison of open source learning management systems. *Procedia-Social and Behavioral Sciences*, 143:521–526, 2014.

[177] KR Premlatha, B Dharani, and TV Geetha. Dynamic learner profiling and automatic learner classification for adaptive e-learning environment. *Interactive Learning Environments*, 24(6):1054–1075, 2016.

[178] Peter Brusilovsky. Adaptive hypermedia for education and training. *Adaptive technologies for training and education*, 46:46–68, 2012.

[179] Andrina Granić and Jelena Nakić. Enhancing the learning experience: Preliminary framework for user individual differences. In *Symposium of the Austrian HCI and Usability Engineering Group*, pages 384–399. Springer, 2010.

[180] L Jegatha Deborah, Ramachandran Baskaran, and Arputharaj Kannan. Learning styles assessment and theoretical origin in an e-learning scenario: a survey. *Artificial Intelligence Review*, 42(4):801–819, 2014.

[181] Suzana Marković and Nenad Jovanović. Learning style as a factor which affects the quality of e-learning. *Artificial Intelligence Review*, 38(4):303–312, 2012.

[182] Slavi Stoyanov and Paul Kirchner. Expert concept mapping method for defining the characteristics of adaptive e-learning: Alfanet project case. *Educational technology research and development*, 52(2):41–54, 2004.

[183] Stephan Weibelzahl. Problems and pitfalls in the evaluation of adaptive systems. In *Adaptable and adaptive hypermedia systems*, pages 285–299. IGI Global, 2005.

[184] Maria Dominic, Anthony Xavier Britto, and Sagayaraj Francis. A framework to formulate adaptivity for adaptive e-learning system using user response theory. *International Journal of Modern Education and Computer Science*, 7(1):23, 2015.

[185] KR Premlatha and TV Geetha. Learning content design and learner adaptation for adaptive e-learning environment: a survey. *Artificial Intelligence Review*, 44(4):443–465, 2015.

[186] Juan E Gilbert and Chia Y Han. Arthur: A personalized instructional system. *Journal of Computing in Higher Education*, 14(1):113–129, 2002.

[187] Kazuya Seki, Tatsunori Matsui, and Toshio Okamoto. An adaptive sequencing method of the learning objects for the e-learning environment. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 88(3):54–71, 2005.

[188] Marcus Specht and Reinhard Oppermann. Ace-adaptive courseware environment. *New Review of Hypermedia and Multimedia*, 4(1):141–161, 1998.

[189] Giuseppe Araniti, Pasquale De Meo, Antonio Iera, and Domenico Ursino. Adaptively controlling the qos of multimedia wireless applications through" user profiling" techniques. *IEEE Journal on Selected Areas in Communications*, 21(10):1546–1556, 2003.

[190] Jeremy E Thompson. Student modeling in an intelligent tutoring system. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH, 1996.

[191] Sabine Graf, Cindy Ives, et al. A flexible mechanism for providing adaptivity based on learning styles in learning management systems. In *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*, pages 30–34. IEEE, 2010.

[192] Martin Weller. Vle 2.0 and future directions in learning environments. In *Proceedings of the first international LAMS conference*, pages 99–106. Citeseer, 2006.

[193] Pithamber R Polsani. Use and abuse of reusable learning objects. *Journal of Digital information*, 3(4), 2003.

[194] Allard Strijker and Betty Collis. *New pedagogies and re-usable learning objects: Toward a different role for an LMS*. Association for the Advancement of Computing in Education (AACE), 2002.

[195] Carla Limongelli, Filippo Sciarrone, Marco Temperini, and Giulia Vaste. Adaptive learning with the ls-plan system: a field evaluation. *IEEE Transactions on Learning Technologies*, 2(3):203–215, 2009.

[196] O Takács. Rule based system of adaptive e-learning. In *2014 IEEE 12th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 465–470. IEEE, 2014.

[197] Sucheta V Kolekar, Radhika M Pai, and Manohara Pai MM. Rule based adaptive user interface for adaptive e-learning system. *Education and Information Technologies*, 24(1):613–641, 2019.

[198] Alexandros Paramythis and Susanne Loidl-Reisinger. Adaptive learning environments and e-learning standards. In *Second european conference on e-learning*, volume 1, pages 369–379, 2003.

[199] FO Isinkaye, YO Folajimi, and BA Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261–273, 2015.

[200] Yinghui Yang and Balaji Padmanabhan. Evaluation of online personalization systems: A survey of evaluation schemes and a knowledge-based approach. *Journal of Electronic Commerce Research*, 6(2):112, 2005.

[201] Paul Thomas and David Hawking. Evaluation by comparing result sets in context. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 94–101. ACM, 2006.

[202] David Elsweiler and Ian Ruthven. Towards task-based personal information management evaluations. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 23–30. ACM, 2007.

[203] Miquel Montaner, B López, and J Ll De la Rosa. A taxonomy of personalized agents on the internet. *Technical Report TR-UDG-2001-05 J. Departament d'Electrònica Informàtica i Automàtica*, 2001.

[204] Amos Azaria, Avinatan Hassidim, Sarit Kraus, Adi Eshkol, Ofer Weintraub, and Irit Netanely. Movie recommender system for profit maximization. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 121–128. ACM, 2013.

[205] Joonseok Lee, Mingxuan Sun, and Guy Lebanon. Prea: Personalized recommendation algorithms toolkit. *Journal of Machine Learning Research*, 13(Sep):2699–2703, 2012.

[206] Michael Jugovac, Dietmar Jannach, and Lukas Lerche. Efficient optimization of multiple recommendation quality factors according to individual user tendencies. *Expert Systems with Applications*, 81:321–331, 2017.

[207] J Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166. ACM, 1999.

[208] Faiçal Azouaou and Cyrille Desmoulins. Using and modeling context with ontology in e-learning: the case of teacher's personal annotation. In *Workshop on Applications of Semantic Web Technologies for e-Learning (SWEL@ AH'06), Dublin, Ireland*, 2006.

[209] Yiouli Kritikou, Panagiotis Demestichas, Evgenia Adamopoulou, Konstantinos Demestichas, Michael Theologou, and Maria Paradia. User profile modeling in the context of web-based learning management systems. *Journal of Network and Computer Applications*, 31(4):603–627, 2008.

[210] Chihiro Ono, Mori Kurokawa, Yoichi Motomura, and Hideki Asoh. A context-aware movie preference model using a bayesian network for recommendation and promotion. In *International Conference on User Modeling*, pages 247–257. Springer, 2007.

[211] Stuart E Middleton, Nigel R Shadbolt, and David C De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):54–88, 2004.

[212] Madura Prabhani Pitigala Liyanage, Lasith Gunawardena KS, and Masahito Hirakawa. Detecting learning styles in learning management systems using data mining. *Journal of Information Processing*, 24(4):740–749, 2016.

[213] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[214] Douglas W Oard, Jinmook Kim, et al. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, volume 83. WoUongong, 1998.

[215] Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier, 1995.

[216] Raymond J Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204. ACM, 2000.

[217] Robin Van Meteren and Maarten Van Someren. Using content-based filtering for recommendation. In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, pages 47–56, 2000.

[218] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. *Aaai/iaai*, 23:187–192, 2002.

[219] Marco Del Tredici and Malvina Nissim. A modular system for rule-based text categorisation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, 2014.

[220] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–59, 1997.

[221] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80, 2003.

[222] Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):13, 2016.

[223] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[224] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, 4(2):133–151, 2001.

[225] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[226] Charu C Aggarwal. Neighborhood-based collaborative filtering. In *Recommender systems*, pages 29–70. Springer, 2016.

[227] Charu C Aggarwal. An introduction to recommender systems. In *Recommender systems*, pages 1–28. Springer, 2016.

[228] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems: introduction and challenges. In *Recommender systems handbook*, pages 1–34. Springer, 2015.

[229] Michael D Ekstrand, F Maxwell Harper, Martijn C Willemsen, and Joseph A Konstan. User perception of differences in recommender algorithms. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 161–168. ACM, 2014.

[230] Charu C Aggarwal. Model-based collaborative filtering. In *Recommender systems*, pages 71–138. Springer, 2016.

[231] Charu C Aggarwal. *Data mining: the textbook*. Springer, 2015.

[232] Siavash Ghodsi Moghaddam and Ali Selamat. A scalable collaborative recommender algorithm based on user density-based clustering. In *The 3rd International Conference on Data Mining and Intelligent Information Technology Applications*, pages 246–249. IEEE, 2011.

[233] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

[234] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

[235] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. Item-based collaborative filtering recommendation algorithms. *Www*, 1:285–295, 2001.

[236] Beitao Guo, Fenglian Qi, and Guangyan Fu. A knowledge-based diagnostic system for pneumatic system. In *2008 International Symposium on Knowledge Acquisition and Modeling*, pages 127–130. IEEE, 2008.

[237] Robin Burke. Hybrid web recommender systems. In *The adaptive web*, pages 377–408. Springer, 2007.

[238] Barry Smyth and Paul Cotter. A personalised tv listings service for the digital tv age. *Knowledge-Based Systems*, 13(2-3):53–59, 2000.

[239] Malak Al-hassan, Haiyan Lu, and Jie Lu. Personalized e-government services: Tourism recommender system framework. In *International Conference on Web Information Systems and Technologies*, pages 173–187. Springer, 2010.

[240] Luca Buriano, Marco Marchetti, Francesca Carmagnola, Federica Cena, Cristina Gena, and Ilaria Torre. The role of ontologies in context-aware recommender systems. In *7th International Conference on Mobile Data Management (MDM'06)*, pages 80–80. IEEE, 2006.

[241] Peretz Shoval, Bracha Shapira, and Veronica Maidel. Ontology-content-based filtering method for personalized newspapers, November 30 2010. US Patent 7,844,592.

[242] Catherine Roussey, Francois Pinet, Myoung Ah Kang, and Oscar Corcho. An introduction to ontologies and ontology engineering. In *Ontologies in Urban development projects*, pages 9–38. Springer, 2011.

[243] Gilles Falquet, Claudine Métral, Jacques Teller, and Christopher Tweed. *Ontologies in urban development projects*. Springer Science & Business Media, 2011.

[244] Marwa Harrathi, Narjess Touzani, and Rafik Braham. A hybrid knowlegde-based approach for recommending massive learning activities. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 49–54. IEEE, 2017.

[245] Salam Fraihat and Qusai Shambour. A framework of semantic recommender system for e-learning. *Journal of Software*, 10(3):317–330, 2015.

[246] John K Tarus, Zhendong Niu, and Abdallah Yousif. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Future Generation Computer Systems*, 72:37–48, 2017.

[247] Dean Allemang and James Hendler. *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier, 2011.

[248] Robin Burke, Bamshad Mobasher, Roman Zabicki, and Runa Bhaumik. Identifying attack models for secure recommendation. *Beyond Personalization*, 2005, 2005.

[249] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. Recommender system application developments: a survey. *Decision Support Systems*, 74:12–32, 2015.

[250] Mark Claypool, Anuja Gokhale, Tim Miranda, Paul Murnikov, Dmitry Netes, and Matthew Sartin. Combing content-based and collaborative filters in an online newspaper. 1999.

[251] Michael J Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, 13(5-6):393–408, 1999.

[252] Philip Lenhart and Daniel Herzog. Combining content-based and collaborative filtering for personalized sports news recommendations. In *CBRecSys@ RecSys*, pages 3–10, 2016.

[253] Shuo Yang, Mohammed Korayem, Khalifeh AlJadda, Trey Grainger, and Sriraam Natarajan. Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive statistical relational learning approach. *Knowledge-Based Systems*, 136:37–45, 2017.

[254] Alejandro Bellogín, Iván Cantador, Fernando Díez, Pablo Castells, and Enrique Chavarriaga. An empirical comparison of social, collaborative filtering, and hybrid recommenders. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):14, 2013.

[255] Tim Miranda, Mark Claypool, Anuja Gokhale, Tim Mir, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *In Proceedings of ACM SIGIR Workshop on Recommender Systems*. Citeseer, 1999.

[256] Mimi M Recker, Andrew Walker, and Kimberly Lawless. What do you recommend? implementation and analyses of collaborative information filtering of web resources for education. *Instructional Science*, 31(4-5):299–316, 2003.

[257] Mahkameh Yaghmaie and Ardeshir Bahreininejad. A context-aware adaptive learning system using agents. *Expert Systems with Applications*, 38(4):3280–3286, 2011.

[258] Nikos Manouselis and Constantina Costopoulou. Experimental analysis of design choices in multiattribute utility collaborative filtering. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(02):311–331, 2007.

[259] Zameer Gulzar, A Anny Leema, and Gerard Deepak. Pcrs: Personalized course recommender system based on hybrid approach. *Procedia Computer Science*, 125:518–524, 2018.

[260] Mohammed E Ibrahim, Yanyan Yang, David L Ndzi, Guangguang Yang, and Murtadha Al-Maliki. Ontology-based personalized course recommendation framework. *IEEE Access*, 7:5180–5199, 2018.

[261] Wei Chen, Zhendong Niu, Xiangyu Zhao, and Yi Li. A hybrid recommendation algorithm adapted in e-learning environments. *World Wide Web*, 17(2):271–284, 2014.

[262] Tiffany Ya Tang and Gordon McCalla. Smart recommendation for an evolving e-learning system. In *Workshop on Technologies for Electronic Documents for Supporting Learning, International Conference on Artificial Intelligence in Education*, pages 699–710, 2003.

[263] Huiyi Tan, Junfei Guo, and Yong Li. E-learning recommendation system. In *2008 International Conference on Computer Science and Software Engineering*, volume 5, pages 430–433. IEEE, 2008.

[264] Ge Liang, Kong Weining, and Luo Junzhou. Courseware recommendation in e-learning system. In *International Conference on Web-based Learning*, pages 10–24. Springer, 2006.

[265] Mohamed Koutheaïr Khribi, Mohamed Jemni, and Olfa Nasraoui. Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. In *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, pages 241–245. IEEE, 2008.

[266] Mojtaba Salehi and Isa Nakhai Kmalabadi. A hybrid attribute–based recommender system for e–learning material recommendation. *IERI Procedia*, 2:565–570, 2012.

[267] Outmane Bourkoukou, Essaid El Bachari, and Mohamed El Adnani. A recommender model in e-learning environment. *Arabian Journal for Science and Engineering*, 42(2):607–617, 2017.

[268] Wenchao Cui, Yi Wang, Yangyu Fan, Yan Feng, and Tao Lei. Localized fcm clustering with spatial information for medical image segmentation and bias field estimation. *Journal of Biomedical Imaging*, 2013:13, 2013.

[269] Jun Ye, Nicole A Lazar, and Yehua Li. Sparse geostatistical analysis in clustering fmri time series. *Journal of neuroscience methods*, 199(2):336–345, 2011.

[270] Lingling An and RW Doerge. Dynamic clustering of gene expression. *ISRN bioinformatics*, 2012, 2012.

[271] Marcilio CP de Souto, Ivan G Costa, Daniel SA de Araujo, Teresa B Ludermir, and Alexander Schliep. Clustering cancer gene expression data: a comparative study. *BMC bioinformatics*, 9(1):497, 2008.

[272] Suresh H Moolgavkar, Roger O McClellan, Anup Dewanji, Jay Turim, E Georg Luebeck, and Melanie Edwards. Time-series analyses of air pollution and mortality in the united states: a subsampling approach. *Environmental health perspectives*, 121(1):73–78, 2012.

[273] Rosaria Ignaccolo, Stefania Ghigo, and Stefano Bande. Functional zoning for air quality. *Environmental and ecological statistics*, 20(1):109–127, 2013.

[274] Wen Shen, Vahan Babushkin, Zeyar Aung, and Wei Lee Woon. An ensemble model for day-ahead electricity demand time series forecasting. In *Proceedings of the fourth international conference on Future energy systems*, pages 51–62. ACM, 2013.

[275] Jarke J Van Wijk and Edward R Van Selow. Cluster and calendar based visualization of time series data. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)*, pages 4–9. IEEE, 1999.

[276] Sunita B Aher and LMR Lobo. Best combination of machine learning algorithms for course recommendation system in e-learning. *International Journal of Computer Applications*, 41(6), 2012.

[277] Paula Andrea RODRÍGUEZ MARÍN, Néstor Duque, and Demetrio Ovalle. Multi-agent system for knowledge-based recommendation of learning objects. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 4(1):80–89, 2015.

[278] Shaimaa M Nafea, Francois Siewe, and Ying He. A novel algorithm for dynamic student profile adaptation based on learning styles. In *Proceedings of SAI Intelligent Systems Conference*, pages 29–51. Springer, 2019.

[279] Mojtaba Salehi. Application of implicit and explicit attribute based collaborative filtering and bide for learning resource recommendation. *Data & Knowledge Engineering*, 87:130–145, 2013.

[280] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):881–892, 2002.

[281] C Romero Morales, AR Porras Pérez, S Ventura Soto, C Hervás Martınez, and A Zafra. Using sequential pattern mining for links recommendation in adaptive hypermedia educational systems. *Current Developments in Technology-Assisted Education*, 2:1016–1020, 2006.

[282] Keith Ying, Maiga Chang, Andrew F Chiarella, Jia-Sheng Heh, et al. Clustering students based on their annotations of a digital text. In *2012 IEEE Fourth International Conference on Technology for Education*, pages 20–25. IEEE, 2012.

[283] Kiran LN Eranki and Kannan M Moudgalya. Evaluation of web based behavioral interventions using spoken tutorials. In *2012 IEEE Fourth International Conference on Technology for Education*, pages 38–45. IEEE, 2012.

[284] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121. ACM, 2005.

[285] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[286] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[287] John Tarus, Zhendong Niu, and Bakhti Khadidja. E-learning recommender system based on collaborative filtering and ontology. *World Academy of Science, Engineering and Technol-*

ogy, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 11(2):256–261, 2017.

[288] Mojtaba Salehi, Isa Nakhai Kmalabadi, and Mohammad Bagher Ghaznavi Ghoushchi. A new recommendation approach based on implicit attributes of learning material. *Ieri Procedia*, 2:571–576, 2012.

[289] Mojtaba Salehi and Isa Nakhai Kmalabadi. Attribute-based recommender system for learning resource by learner preference tree. In *Computer and Knowledge Engineering (ICCKE), 2012 2nd International eConference*, pages 133–138. IEEE, 2012.

[290] Amer Al-Badarenah and Jamal Alsakran. An automated recommender system for course selection. *International Journal of Advanced Computer Science and Applications*, 7(3):1166–1175, 2016.

[291] Shanshan Wan and Zhendong Niu. An e-learning recommendation approach based on the self-organization of learning resource. *Knowledge-Based Systems*, 160:71–87, 2018.

[292] Yu Zhu, Jiong Mu, Qiang Tang, Jitao Wang, Haitao Li, Zhili Li, Xiaoqing Wang, Hui Yang, and Yinjun Gao. Application of intelligent course selection recommendation system based on ipv6. In *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pages 1–5. IEEE, 2018.

[293] Xiaoxian Huang, Yong Tang, Rong Qu, Chunying Li, Chengzhe Yuan, Saimei Sun, and Bixia Xu. Course recommendation model in academic social networks based on association rules and multi-similarity. In *2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD))*, pages 277–282. IEEE, 2018.

[294] Kiratijuta Bhumichitr, Songsak Channarukul, Nattachai Saejiem, Rachsuda Jiamthapthaksin, and Kwankamol Nongpong. Recommender systems for university elective course recommendation. In *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–5. IEEE, 2017.

[295] Swapaka Listya Trusthi and Dade Nurjanah. Combination of hybrid filtering and learning style for learning material recommendation. In *2017 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, pages 24–29. IEEE, 2017.

[296] Per-Erik Danielsson. Euclidean distance mapping. *Computer Graphics and image processing*, 14(3):227–248, 1980.

[297] Min Ji, Fuding Xie, and Yu Ping. A dynamic fuzzy cluster algorithm for time series. In *Abstract and Applied Analysis*, volume 2013. Hindawi, 2013.

[298] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[299] Pierre Legendre and Loic FJ Legendre. *Numerical ecology*, volume 24. Elsevier, 2012.

[300] Joseph Lee Rodgers and W Alan Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[301] Daxin Jiang, Chun Tang, and Aidong Zhang. Cluster analysis for gene expression data: a survey. *IEEE Transactions on Knowledge & Data Engineering*, (11):1370–1386, 2004.

[302] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[303] Carol Dulong. Apparatus and method for determining the manhattan distance between two points, January 24 1995. US Patent 5,384,722.

[304] Shaimaa Nafea, Francois Siewe, and Ying He. Ulearn: Personalised learner's profile based on dynamic learning style questionnaire. In *Proceedings of Intelligent Systems(IntelliSys)*, pages 1257–1264. IEEE, 2018.

[305] Xia Ning, Christian Desrosiers, and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 37–76. Springer, 2015.

[306] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.

[307] Badrul M Sarwar, George Karypis, Joseph Konstan, and John Riedl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, volume 1, pages 291–324, 2002.

[308] Hyung Jun Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, 178(1):37–51, 2008.

[309] Jesús Bobadilla, Fernando Ortega, and Antonio Hernando. A collaborative filtering similarity measure based on singularities. *Information Processing & Management*, 48(2):204–217, 2012.

[310] Deepa Anand and Kamal K Bharadwaj. Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities. *Expert systems with applications*, 38(5):5101–5109, 2011.

[311] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[312] Sandeep Chanda and Damien Foggon. Entity framework data access paradigms. In *Beginning ASP. NET 4.5 Databases*, pages 137–156. Springer, 2013.

[313] John D Gould and Clayton Lewis. Designing for usability: key principles and what designers think. *Communications of the ACM*, 28(3):300–311, 1985.

[314] José M Aguilar et al. *SignalR programming in Microsoft ASP. NET*. Microsoft Press, 2014.

[315] Jean-Louis Boulanger. *Static analysis of software: The abstract interpretation*. John Wiley & Sons, 2013.

[316] Tafline Murnane and Karl Reed. On the effectiveness of mutation analysis as a black box testing technique. In *Proceedings 2001 Australian Software Engineering Conference*, pages 12–20. IEEE, 2001.

[317] BSEN ISO and BRITISH STANDARD. Ergonomics of human-system interaction. 2010.

[318] Yi-Shun Wang. Assessment of learner satisfaction with asynchronous electronic learning systems. *Information & Management*, 41(1):75–86, 2003.

[319] Tzone-I Wang, Kun-Te Wang, and Yueh-Min Huang. Using a style-based ant colony system for adaptive learning. *Expert Systems with applications*, 34(4):2449–2464, 2008.

[320] Daniel Y Shee and Yi-Shun Wang. Multi-criteria evaluation of the web-based e-learning system: A methodology based on learner satisfaction and its applications. *Computers & Education*, 50(3):894–905, 2008.

[321] Pei-Chen Sun, Ray J Tsai, Glenn Finger, Yueh-Yang Chen, and Dowming Yeh. What drives a successful e-learning? an empirical investigation of the critical factors influencing learner satisfaction. *Computers & education*, 50(4):1183–1202, 2008.

[322] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[323] James R Lewis and Jeff Sauro. The factor structure of the system usability scale. In *International conference on human centered design*, pages 94–103. Springer, 2009.

[324] Konstantina Orfanou, Nikolaos Tselios, and Christos Katsanos. Perceived usability evaluation of learning management systems: Empirical evaluation of the system usability scale. *International Review of Research in Open and Distributed Learning*, 16(2):227–246, 2015.

[325] Carlo De Medio, Carla Limongelli, Filippo Sciarrone, and Marco Temperini. Moodlerec: A recommendation system for creating courses using the moodle e-learning platform. *Computers in Human Behavior*, 104:106168, 2020.

[326] Melody Siadaty and Fattaneh Taghiyareh. Pals2: Pedagogically adaptive learning system based on learning styles. In *Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)*, pages 616–618. IEEE, 2007.

[327] Shahida M Parvez and Glenn D Blank. A pedagogical framework to integrate learning style into intelligent tutoring systems. *Journal of Computing Sciences in Colleges*, 22(3):183–189, 2007.

[328] Karim Moharm. A framework for adaptive personalized e-learning recommender systems. *International Journal of Intelligent Information Systems*, 8(1):12, 2019.

[329] Eiman Aeiad et al. *A framework for an adaptable and personalised e-learning system based on free web resources*. PhD thesis, University of Salford, 2017.

[330] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[331] Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*, 2009.

[332] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.

[333] Oliver Oechslein and Thomas Hess. Incorporating social networking information in recommender systems: The development of a classification framework. In *Bled eConference*, page 19, 2013.

Appendix A:The FSLSM Learning-Style Questionnaire

This version of the FSLSM Questionnaire available at:

http://www.engr.ncsu.edu/learningstyles/ilsweb.html.

Question 1: I understand something better after I

(a) try it out.

(b) think it through.

Question 2: I would rather be considered as

(a) realistic.

(b) innovative.

Question 3: When I think about what I did yesterday, I am most likely to get

(a) a picture.

(b) words.

Question 4: I tend to

(a) understand details of a subject but may be fuzzy about its overall structure.

(b) understand the overall structure but may be fuzzy about details.

Question 5: When I am learning something new, it helps me to

(a) talk about it.

(b) think about it.

Question 6: If I were a teacher, I would rather teach a course

(a) that deals with facts and real life situations.

(b) that deals with ideas and theories.

Question 7: I prefer to get new information in

(a) that deals with facts and real life situations.

(b) that deals with ideas and theories.

Question 8: Once I understand

(a) all the parts, I understand the whole thing.

(b) the whole thing, I see how the parts fit.

Question 9: In a study group working on difficult material, I am more likely to

(a) jump in and contribute ideas.

(b) sit back and listen.

Question 10: I find it easier

(a) to learn facts.

(b) to learn concepts.

Question 11: In a book with lots of pictures and charts, I am likely to

(a) look over the pictures and charts carefully.

(b) focus on the written text..

Question 12: When I solve math problems

(a) I usually work my way to the solutions one step at a time.

(b) I often just see the solutions but then have to struggle to figure out the steps to get to them.

Question 13: In classes I have taken

(a) I have usually gotten to know many of the students.

(b) I have rarely gotten to know many of the students.

Question 14: In reading nonfiction, I prefer

(a) something that teaches me new facts or tells me how to do something.

(b) something that gives me new ideas to think about.

Question 15: I like teachers

(a) who put a lot of diagrams on the board.

(b) who spend a lot of time explaining.

Question 16: When I am analyzing a story or a novel

(a) I think of the incidents and try to put them together to figure out the themes.

(b) I just know what the themes are when I finish reading and then I have to go back and find the incidents that demonstrate them.

Question 17: When I start a homework problem, I am more likely to

(a) start working on the solution immediately.

(b) try to fully understand the problem first.

Question 18: I prefer the idea of

(a) certainty.

(b) theory.

Question 19: I remember best

(a) what I see.

(b) what I hear.

Question 20: It is more important to me that an instructor

(a) lays out the material in clear sequential steps.

(b) gives me an overall picture and relates the material to other subjects.

Question 21: I prefer to study

(a) in a study group.

(b) alone.

Question 22: I am more likely to be considered as

(a) careful about the details of my work.

(b) creative about how to do my work.

Question 23: When I get directions to a new place, I prefer

(a) a map.

(b) written instructions.

Question 24: I learn

(a) at a fairly regular pace. If I study hard, I will “get it”.

(b) in fits and starts. I will be totally confused and then suddenly it all “clicks”.

Question 25: I would rather first

(a) try things out.

(b) think about how I am going to do it.

Question 26: When I am reading for enjoyment, I like writers to

(a) clearly say what they mean.

(b) say things in creative, interesting ways.

Question 27: When I see a diagram or sketch in class, I am most likely to remember

(a) the picture.

(b) what the instructor said about it.

Question 28: When considering a body of information, I am more likely to

(a) focus on details and miss the big picture.

(b) try to understand the big picture before getting into the details.

Question 29: I more easily remember

(a) something I have done.

(b) something I have thought a lot about.

Question 30: When I have to perform a task, I prefer to

(a) master one-way of doing it.

(b) come up with new ways of doing it.

Question 31: When someone is showing me data, I prefer

(a) charts or graphs.

(b) text summarizing the results.

Question 32: When writing a paper, I am more likely to

(a) work on (think about or write) the beginning of the paper and progress forward.

(b) work on (think about or write) different parts of the paper and then order them.

Question 33: When I have to work on a group project, I first want to

(a) have "group brainstorming" where everyone contributes ideas.

(b) brainstorm individually and then come together as a group to compare ideas.

Question 34: I consider it higher praise to call someone as

(a) sensible.

(b) imaginative.

Question 35: When I meet people at a party, I am more likely to remember

(a) what they looked like.

(b) what they said about themselves.

Question 36: When I am learning a new subject, I prefer to

(a) stay focused on that subject, learning as much about it as I can.

(b) try to make connections between that subject and related subjects.

Question 37: I am more likely to be considered as

(a) outgoing.

(b) reserved.

Question 38: I prefer courses that emphasize

(a) concrete material (facts, data).

(b) abstract material (concepts, theories).

Question 39: For entertainment, I would rather

(a) watch television.

(b) read a book.

Question 40: Some teachers start their lectures with an outline of what they will cover. Such outlines are

(a) somewhat helpful to me.

(b) very helpful to me.

Question 41: The idea of doing homework in groups, with one grade for the entire group,

(a) appeals to me.

(b) does not appeal to me.

Question 42: When I am doing long calculations

(a) I tend to repeat all my steps and check my work carefully.

(b) I find checking my work tiresome and have to force myself to do it.

Question 43: I tend to picture places I have been

(a) easily and fairly accurately.

(b) with difficulty and without much detail.

Question 44: When solving problems in a group, I would be more likely to

(a) think of the steps in the solution process.

(b) think of possible consequences or applications of the solution in a wide range of areas.

Appendix B:ULEARN Satisfaction Questionnaire.

In order to evaluate student satisfaction with the ULEARN system, we conducted a user-centred evaluation using E-Learner satisfaction questionnaire [318] and System Usability Scale questionnaire [322].

Dear participants

Please mark your response from questions 1 to 8 by clicking ✓ for each question. There are no or right or wrong answers to the questions in this questionnaire. Select the most appropriate answer for each question-based in your view/experience. Thank you for your participation in this study.

	Strongly disagree				Strongly agree
1. This system makes it easy for you to find the content you need.	1	2	3	4	5
2. This system provides content that exactly fits your needs, and I did not have to look for them in the entire course.	1	2	3	4	5
3. This system provides sufficient content.	1	2	3	4	5
4. This system makes it easy for you to discuss questions with other students.	1	2	3	4	5
5. It was easier to solve my difficulties in understanding the topic with the help of the recommendations of this system.	1	2	3	4	5
6. I feel I learn more while using this system.	1	2	3	4	5
7. I believe I became productive using this system and its promote my learning interests.	1	2	3	4	5
8. I am satisfied with the quality of the personalisation..	1	2	3	4	5
9. Overall, I am satisfied with this system.	1	2	3	4	5
10. I would recommend this system to my classmates	1	2	3	4	5

Appendix C: ULEARN System Usability Questionnaire.

The questions of the System Usability Scale [322].

		Strongly disagree				Strongly agree
11.	I think that I would like to use ULEARN frequently.	1	2	3	4	5
12.	I found the ULEARN system unnecessarily complex.	1	2	3	4	5
13.	thought the ULEARN system was easy to use.	1	2	3	4	5
14.	I think that I would need the support of a technical person to be able to use this system.	1	2	3	4	5
15.	I found the various functions in this system were well-integrated.	1	2	3	4	5
16.	thought there was too much inconsistency in this system.	1	2	3	4	5
17.	I would imagine that most students would learn to use this system very quickly.	1	2	3	4	5
18.	I found the ULERAN recommender system very difficult to use.	1	2	3	4	5
19.	I felt very confident using this system.	1	2	3	4	5
20.	I need to learn a lot of things before I could get going with this system.	1	2	3	4	5

Appendix D: ULEARN system interface pages

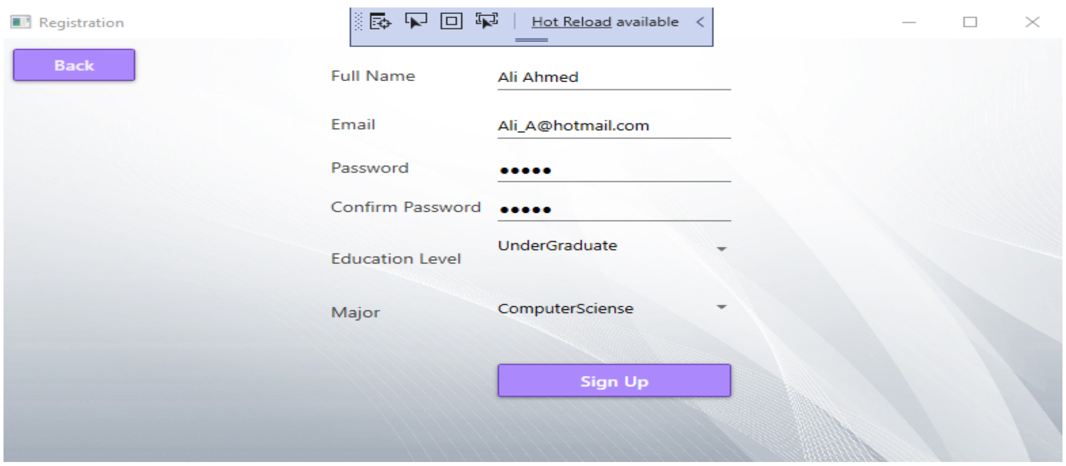


Figure D.1: Student registration page in ULEARN system

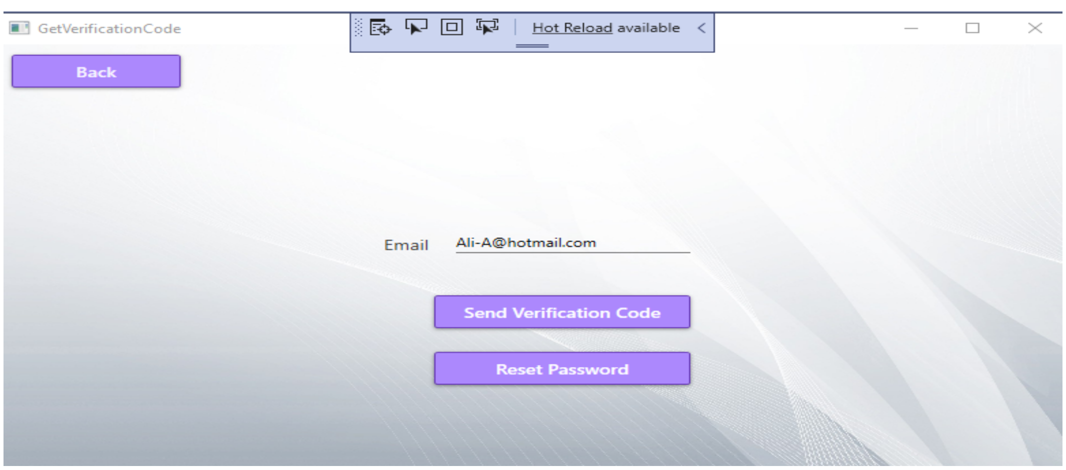


Figure D.2: Registration page forgot password

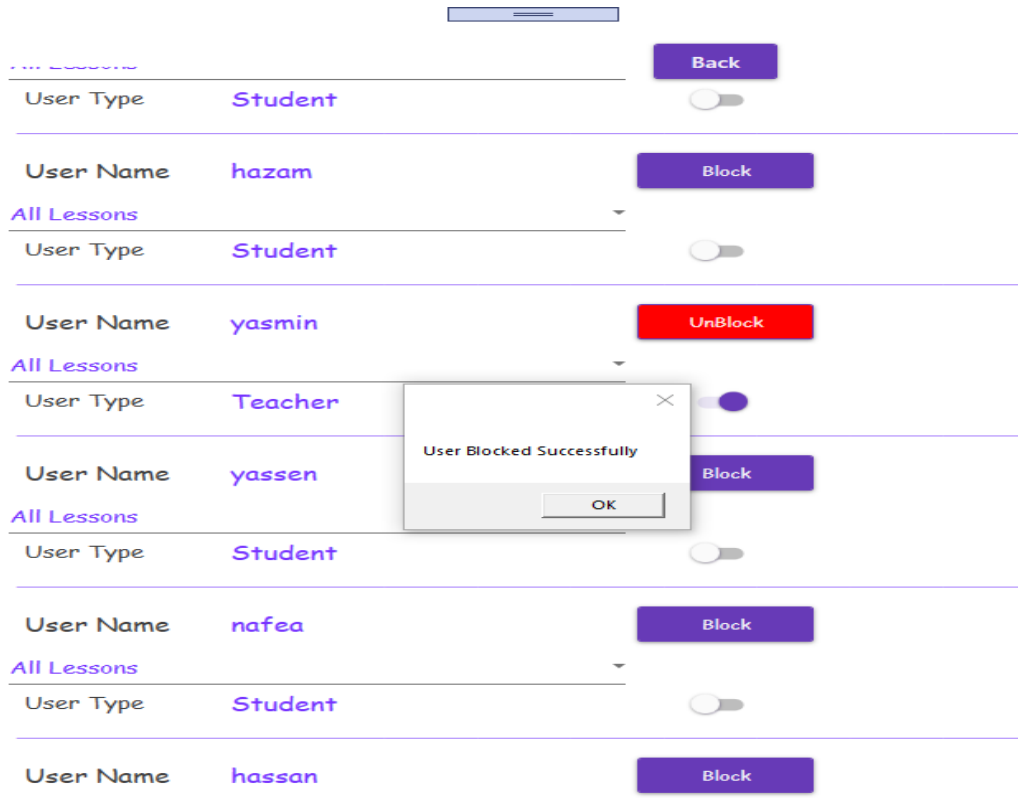


Figure D.3: Administrator's page

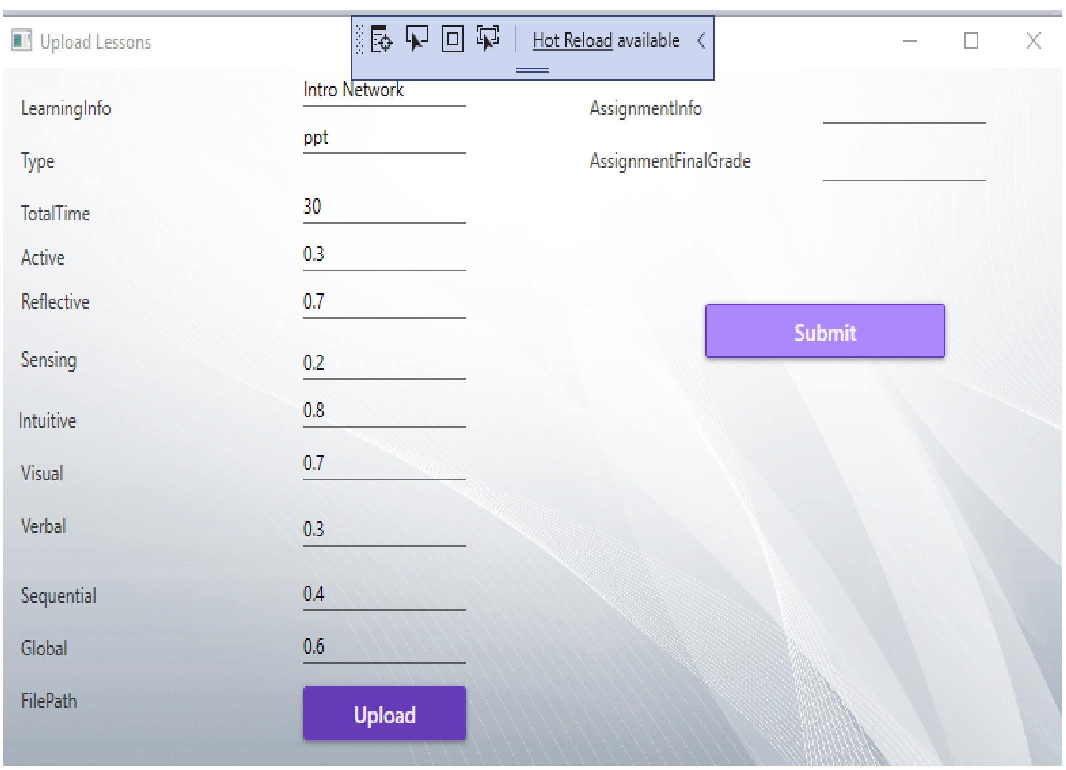


Figure D.4: Teacher's page (learning object profile creation)

Appendix E: ULEARN system Test Cases

Table E.1: Test cases for teacher interface

Test Case Name: Teacher portal				
Test Case ID	**Test Case Scenario**	**Test Case**	**Expected Results**	**Status (Pass /Fail)**
1	Upload learning object	Test to see if teacher can upload learning objects.	Learning object uploaded.	Pass
2	Update learning objects	Delete and update learning objects files.	learning objects deleted and updated	Pass
3	learning object profile	Test to see if teacher can create learning object profile	learning objects profile created	Pass
4	Assignment	Test to see if teacher can add assignment.	The expected result is the assignment is created.	pass
5	Assignment's Questions	Test to see if teacher can add questions to assignment.	The expected result is that the assignment questions are loaded when add to assignment button is pressed.	Pass
6	Send messages	Test to see if teacher can send messages to students.	The expected result is that messages will appear on a group chat.	Pass

Table E.2: Test cases for Admin interface

Test Case Name: Administrator portal				
Test Case ID	**Test Case Scenario**	**Test Case**	**Expected Results**	**Status (Pass /Fail)**
1	Add Teacher	Test to see if admin can add teacher information.	The expected result is that teacher profile saved in ULEARN database.	Pass
2	Block user	Test to see if admin can block user account.	The expected result is that block button works and leads to blocking user account.	Pass
3	Assign course to teachers	Test to see if admin can assign teacher to course.	The expected result is that course added to teacher portal.	Pass
4	Access database	Test to see if admin can update and delete courses from the database.	The expected result is that course updated in the system database.	Pass